FUZZY LOGIC CLASSIFICATION OF HANDWRITTEN SIGNATURE BASED COMPUTER ACCESS AND FILE ENCRYPTION.

BY

Emmanuel Kwarteng

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in
Computer and Information Science

University of Michigan – Flint, 2010

Flint, MI

Approved by:

Thesis Advisor (Dr. Michael E. Farmer)

Stephen W Turner

Second Reader (Dr. Stephen W. Turner)

ABSTRACT

Often times computer access and file encryption is successful based on how complex a password will be, how often users could change their complex password, the length of the complex password and how creative users are in creating a complex password to stand against unauthorized access to computer resources or files. This research proposes a new way of computer access and file encryption based on the fuzzy logic classification of handwritten signatures. Feature extraction of the handwritten signatures, the Fourier transformation algorithm and the k-Nearest Neighbor Algorithm could be implemented to determine how close the signature is to the signature on file to grant or deny users access to computer resources and encrypted files. Alternatively implementing fuzzy logic algorithms and fuzzy k-Nearest Neighbor algorithm to the captured signature could determine how close a signature is to the one on file to grant or deny access to computer resources and files. This research paper accomplishes the feature recognition firstly by extracting the features as users sign their signatures for storage, and secondly by determining the shortest distance between the signatures. On the other hand this research work accomplish the fuzzy logic recognition firstly by classifying the signature into a membership groups based on their degree of membership and secondly by determining what level of closeness the signatures are from each other. The signatures were collected from three selected input devices- the mouse, I-Pen and the IOGear. This research demonstrates which input device users found efficient and flexible to sign their respective names. The research work also demonstrates the security levels of implementing the fuzzy logic, fuzzy k-Nearest Neighbor, Fourier Transform.

DEDICATION

This research work is dedicated to the Almighty God, the giver of life and the sustainer of humanity for giving me knowledge, wisdom, strength and power to carry on this research paper. The next dedication goes to my wife, Jamila Kwarteng for her support since the beginning of this research work. This research work is equally dedicated to my parents Mr. Timothy Kwabena Ntiamoah and Mrs. Grace Ntiamoah, my guardian Mr. Tabi Kwabena Gyansah and Mrs. Mary Gyansah for instilling within me the thirst for knowledge and the quest for excellence. This research work is highly dedicated to my immediate and extended family for rallying behind me and contributing to my education both in cash and in kind. Finally, I want to dedicate this research work to my professors, friends and church family who have all taught and inspired me to reach higher heights. May the good Lord bless each and every one in the worthy name of Jesus. Amen!

ACKNOWLEDGEMENTS

This research work will not have been possible without the research efforts of many individuals in the fuzzy logic field of research. This main idea of fuzzy was first introduced by Professor Lofti A. Zadeh in 1965 and this concept has revolutionalized the logic of mathematical modeling, computing and analyzing information which rises from human cognition. We pause to share gratitude to Professor Lofti A. Zadeh and individuals who have expanded his ideas over decades now.

Secondly, my acknowledgement goes to Dr. Michael E. Farmer, Ph.D. Associate Professor of Computer Science, in Computer Science, Engineering Science and Physics Department (CSESP) Graduate Program Director of Computer and Information Systems at the University of Michigan Flint campus. His unfailing support has contributed to the success of this research work. I acknowledge the review board for their constructive criticism that has seen this research paper reach this far.

I also want to acknowledge the stimulating and intellectual environment that the UMFLINT Office of Graduate Programs provided during the course of this research work. This inspiring environment has been largely responsible for the accomplishment of this research work by granting me a Graduate Research Assistantship Position to facilitate the completion of this research work.

I sincerely acknowledge all those who took some time out of their busy schedule to sign their various names for the purpose of this research work. My acknowledgement goes to Ashwin Mahajan, Dr. R. Shantaram and Sue Helser for their contribution towards the success of this research paper.

TABLE OF CONTENTS

1. INTRODUCTION

There are many issues that arise when it comes to accessing of private data or information within this era where many fraud and unauthorized access to files and documents are at its peak. There are many research efforts in this field aiming to provide the best, easiest, and most secure means of accessing important data on a drive such as one's legal information, social security numbers, credit card information et cetera. Currently, different levels of password have been implemented for accessing data and also preventing unauthorized data/file access. There are many policies that go along with creating and maintaining a good but complex password. Users have to create a complex password in order to make the work of the hacker difficult. Users are required to change their password often because passwords could be hacked with the test of time. Users are recommended every now and then to create complex password in order to make the password less vulnerable. Unfortunately, users often times forget about the complex password they have created. Users who are nervous about forgetting their passwords are forced to write it down making the password vulnerable.

After several studies and research, there was the need to create more advanced way of protecting user's information and a system that will relieve users of the stress of creating complex passwords, changing passwords often and at the same time being able to remember those passwords. These studies gave birth to the fingerprint technology. Although there are numerous advantages that come with the implementation of the fingerprint technology, there are few disadvantages that worth an alternative way of accessing our documents and files sitting on our computers. According to Jamieson, et al. [2], user acceptance has been the major drawback of this technology. Temporal and permanent injury can also interfere with the scanning process which affects the implementation of this technology. Finally, secured ways of storing, protecting and maintaining these vital data is crucial.

This paper will be presenting an alternative for computer access and file encryption using the handwritten signature. Handwritten signature is a unique way of authenticating documents which cut across all sectors of our day to day activities such as bank transactions, withdrawing money and cashing cheques from the bank, legal documents, certificates and the like. With handwritten signature, users do not have to worry about;

1. The Complexity of their password.
2. How often they have to change their password.
3. How diligent they have to be in order to protect their documents.
4. Keeping and remembering their complex passwords.

Since signing of signatures has been part of the daily activities of users, it is assumed that most users will find handwritten signature as the most appropriate, easy to use, reliable and authentic method of protecting and getting access to computers. In addition, users do not have to worry about hackers hacking their password. Finally, handwritten signature will be widely accepted since it does not infringe on the privacy of users as in the case of fingerprints.

1.1 RESEARCH PROBLEM DESCRIPTION

This paper is aimed at classifying and analyzing how handwritten signature could be used to allow users access to their computers and encrypted files using fuzzy logic. To arrive at this goal analytical experiments were conducted which paved way to test the research ideas on well researched algorithms. Handwritten signatures were captured with three different input devices using a capturing tool that was designed using MATLAB. The three input devices used for capturing the handwritten signatures were an ordinary mouse, an I-Pen and an IOGear.

The mouse, the I-pen and a Bluetooth-enable pen (IOGear) were selected to facilitate this achievement. To generate a unique and more accurate handwritten signature, users should be able to use writing materials or tools that are readily available, frequently used, universally accepted and flexible. A conclusion will be drawn as to which input device users find more efficient and much easier as compared to other input devices and how they adapted to the use of the input device as they sign their names multiple times. The Mouse was one of the input devices because the computer mouse is one of the main input devices for most of the desktop computers. The computer mouse is widely accepted across the globe where computers are most often used and forms part of the daily activities. I-Pen and the IOGear were also considered as input devices for the purpose of this research because they exhibit a pen like behavior which is assumed to be easy for users to adapt since most users use pen most of the time in documenting data/information for their record keeping.

Users might have been exposed to the technology behind these input devices at least once. These input devices will be analyzed based on how comfortable the input devices are to the users and availability of the input devices. These among others will help determine the right choice of input device among the three selected input devices.

The approach to achieving the goal of classifying handwritten signatures for computer access and file encryption is a fuzzy based approach. Many people go by writing almost different signatures each time they sign their name. There is always uncertainty in their handwritten signature. Fuzzy logic based classification will aim at classifying the uncertain handwritten signature based on a well researched algorithms. Fuzzy logic, which is derived from fuzzy set theory, is a branch of mathematics which deals with reasoning that is approximate rather than exact. With this approach the variable that will be used will be referred to as the membership values. Numeric values will be assigned to these variables just as fuzzy set theory. The numeric values will range exclusively between 1 and 0 representing degree of similarity of the condition. Section 2 of this paper delves more into the details of how the fuzzy logic works, and in Section 3 we will implement the ideas established during the background review in Section 2.

1.2 OVERVIEW OF THE THESIS

Electronic tablets accurately capture the *x* and *y* coordinates of the pen tip movement during writing. The advent of these tablets during the mid 1950's was instrumental in on-line handwriting recognition. This technology has lasted until recently and it has been the basis for the evolvement of new technologies. Over the years there has been a renewal of how handwritten data are recognized based on different forms of factors. There has been less error in terms of the recognition of hand written data as compared to early stages of this technology. The hardware advance of combining tablets and flat displays brings input and output into the same surface [46]. As time goes by, the understanding of the requirement of appropriate and more accurate way of recognizing handwritten signatures, text, symbols and characters advances. Amongst these new ideas that could potentially classify and analyze handwritten signatures is fuzzy logic.

Throughout this paper, ideas from prior research works which implemented fuzzy logic in string matching, handwritten document classification, signature recognition, signature authentication etc. were reviewed. Alternatively, this paper reviewed other papers which implement other ideas of uncertainties. The best, simplest and well accepted algorithms were adapted for the purpose of this study and tools were designed with MATLAB to facilitate the collection, studying and analyzing of the data.

1.3 THE OUTLINE OF THE THESIS

This paper is divided into five Sections, the introduction, related work, details of the research, experimental evaluation and conclusion. The first section which is labeled Section 1 will cover the introduction of the paper. This section was further divided to take care of the problem description, the overview of the thesis and the outline of the thesis. Section 2 was devoted to look at other related work done in previous years. This section is labeled as related work. Some of the topics of concern are Overview of Signature Recognition, Offline versus Online Signature Recognition, Online Handwritten Properties and Recognition Problems, Overview of Processing required for Signature Based Access, Preprocessing of Handwritten Signatures, Feature Extraction, Fractal

Features, Fourier Descriptors, Classification and Decision Making, Feature-Based Recognition Methods, String-Based Recognition Methods, Dynamic Programming, Dynamic Time Warping, Review of Fuzzy Logic and Evaluating signature verification Systems. After investigating other related work, the following Section labeled 'Details of Research' will discuss the Technical Details of the research work followed by Input Devices, Signature Capture, the Layout of the signature, Storage and Retrieval of the signatures, Feature extraction, a detailed work on Fuzzy Classification and the Fuzzy Nearest Neighbor algorithms.

Section 4 will deal with the evaluation of the research. It is labeled as Experimental Evaluation. In this section the implementation of the accumulated ideas, Methodology used, the outline of the Test Procedures, the outcome of the experiment and the recommendations were the main focus. The last section will be the conclusion and future works. The summary of what the previous four sections have been discussing and the future directions discovered along this research was well documented in this section. This section is followed up with additional tables, figures and references.

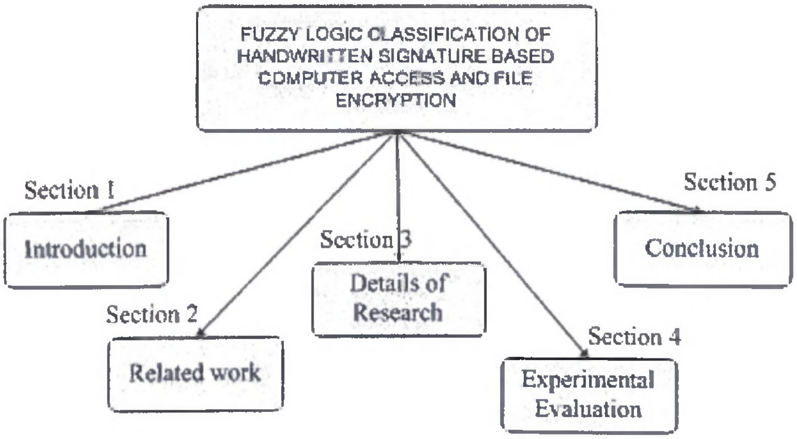


Figure 1.3.1 The major sections of the paper

2. RELATED WORK

2.1 OVERVIEW OF SIGNATURE RECOGNITION

Machine recognition of signature is a very special and difficult problem. Those constraints arise due to the complexity of signature patterns, associated large variations in the patterns of a single person's signature and the forged signatures produced by professional forgers [25].

The major objective of signature recognition is to identify the writer who wrote that signature and the identification process relies upon verification which confirms or rejects the sample [14]. The term verification is encountered most frequently in the context of signatures [27]. Handwritten signatures are commonly used to approve or authorize content of a document or to authenticate financial transactions. Handwritten signature verification is often accomplished by visually inspecting the signed signature. The author of the signature compares the appearance of the sampled signature and either accepts or rejects the signature if it is sufficiently similar to the referenced signature [72]. On-line signature verification scheme aims to extract signature features that reflect the spatial and the temporal characteristics of a signature [56] [72]. Many forms of signature verifications have been proposed. In reference to [54], [55], signature verification is based on the notion that, signatures are produced by ballistic motions. That is motion that does not require any visual feedback and they are difficult to mimic because they are produced naturally. The ballistic motion could easily be captured from how the tip of the pen moves, the speed at which it moves and the kind of force that it generates [46].

The main problem facing signature verification is professional forgers. A professionally forged signature cannot be easily spotted except with a careful trained human eye. The force of the writer, his speed and pressure while signing are proved to be almost unique and difficult to mimic. Hence, on-line dynamic techniques were more successful. [57]

There are numerous taxonomies with which signature recognition can be divided. A key discriminator approach is based on the method of data acquisition. How data is collected determines the classification/group of signature recognition to which it belongs. The two main categories of signature recognition are on-line and off-line signature recognition, as described in [15], [16]. With off-line signature recognition, the signature is available on paper or any other form of writing material from which it will be scanned to get the digital representation of the entire original signature as shown in Figure 2.1.1.1a below. With on-line signature recognition, signature is captured in real-time with a specialized tools/hardware. Both on-line and off-line methods of capturing handwritten signature have their own advantages and disadvantages, and we will discuss each of them in detail in the following section. Note, however, for the purpose of this research, the mouse, Bluetooth enabled pen (IOGear) and an I-pen will be used. These capturing devices are specially designed to digitize the signature during the capturing process which means our basic approach will be an on-line approach.

2.1.1 OFFLINE VERSUS ONLINE SIGNATURE RECOGNITION

The two main categories of signature recognition are on-line and off-line signature recognition, as described in [15], [16]. With off-line signature recognition, the signature is available on paper or any other form of writing material. The signature is then scanned to get the digital representation of the original signature as shown in Figure 2.1.1.1a below. With on-line signature recognition, signature is captured in real-time as the person signs with a specialized tools/hardware. Example of handwritten online signature is as shown in Figure 2.1.1.1b below.

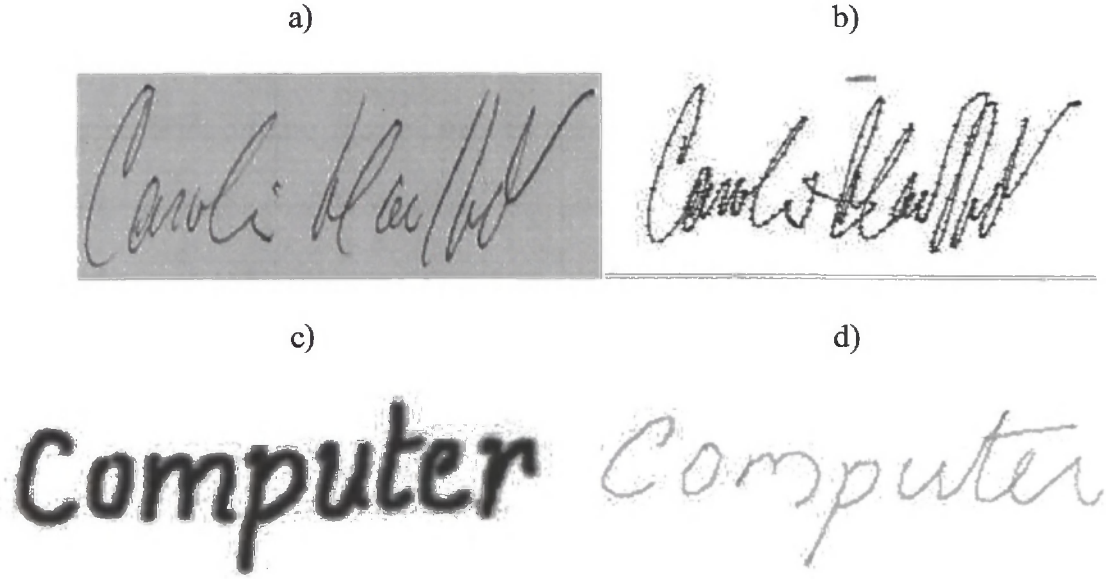


Figure 2.1.1.1 samples of handwritten signatures a) Off-line handwritten signature b) On-line hand written signature c) Off-line handwritten word d) On-line handwritten word

The off-line data is easy to acquire since you only have to get paper and a pen to be in the position to sign. The only problem we will encounter with off-line is during the conversion process when the handwritten signature needs to be digitized. The hardware device required to facilitate the conversion process is the scanner. Past signatures collected over the years could be converted into a digitized format using the off-line handwritten signature collections. This approach is not attractive for the purpose of this research because once a hacker gains access to someone's signature on a sheet of paper, they would continue to have access to owner's data and system. The processing steps needed to extract important features from the signature for recognition are more difficult. There isn't much information available to indicate the sequence of processes in which the strokes were formed. Therefore all the features have to be extracted from the digitized signature pattern [16], [17]. The on-line signature recognition uses the dynamics of hand movements of the signature in addition to its shape. With the online capturing method the individual has to be present at the time of signing. The on-line signature recognition requires a special tool/hardware to be installed. This can be an obstacle for potential customers. With on-line signature recognition, extra characteristic features such as time dependencies and possible pressure and pen tilt which are useful for recognition are captured as well [18].

The online recognition systems can acquire time dependent information like acceleration of the signing and the applied pressure during the writing. Therefore the online recognition systems provide excellent recognition rates. The most important characteristic of online documents is ability to capture the temporal sequence of the stroke while writing [3]. This allows each individual strokes to be analyzed easily. Additional temporal information can be used for document understanding and pattern recognition. The online signature recognition system can capture feature that can be instrumental in terms of determining the writer of the signature. Identifying the right owner of a handwritten signature sometimes becomes difficult but it could be made simpler if other dependant information such as the acceleration of the writing, the amount of pressure applied during the writing of the signature etc. is captured as well. On-line systems generally present a better performance than the off-line method of capturing signature. The on-line method requires the presence of the writer during both the acquisition process and the verification process [15], [27], [28]. Although the off-line method of capturing handwritten signature generally do not require any specialized and complex hardware rather than the scanner, it requires a complex preprocessing steps which also end up generating a huge size of the database [15], [27], [28]. Online data provides temporal information which can be distinctly valuable in several situations [68].

To deal with the difficulties that both on-line and off-line handwritten signature capturing brings, A. Zimmer and L. Ling in their paper propose a hybrid on/off line handwritten signature verification. Their approach was to limit the use of digitizing tablets in acquisition of reference data while carrying out a verification process which could be done directly over a desired document with the presence of the writer of the signature [16]. To achieve this, the online reference data should be used as the basis for the localized feature extraction process while the off-line data should be segmented during the verification process [16].

The hybrid architecture of capturing handwritten signatures was divided into two modules namely the acquisition and training module and the verification module [16]. The first stage is a combination of both on-line and off-line and the later stage is the verification process, which is done off-line. At the first stage the on-line data is captured and processed and all the thresholds needed for the verification process is also generated. During the last stage, test image is introduced into the verification system where the similarities between the reference data and the test data are extracted followed by the authenticity of the signature.

2.1.2 ONLINE HANDWRITTEN PROPERTIES AND RECOGNITION PROBLEMS

A written language has an alphabet of characters/letters, punctuations, symbols and many others. The main property of these letters/characters, symbols and punctuations that make communication possible is the difference identified among them. The hand writing consists of a sequence of strokes [46], which concludes that, all handwritten signatures are sequence of strokes. This is identified in writing when letters/characters follow each other in a specific order. Considering the English forms of writing, the language has upper and lower cases for all the 26 characters/letters and basically two style of writing (cursive and printing). The positions as well as the size of English letters are crucial during the writing and recognition of characters. Lower case letter are small in size as compared to the upper case letter which are of a full size and sit on the baseline. Some of the lower case letters ascend to the height of the uppercase letters and some of the lower case letters descend below the baseline. All characters/letters vary in both static and dynamic properties [46].

Static variation can occur in shape or size and dynamic variations could also occur in stroke number and order. English language writings might have more variations in stroke direction. These variations are more dependent on the presence or absence of retraces (overwriting of stroke). The variations which occur are more dependent on the style and the speed at which the author writes. Variability of handwriting and recognition has been well documented by [47], [48], [49], and [50].

There are flaws, errors and problems associated with online handwritten recognition due to the fact that many people tend to write the same English language letters and symbols in different ways [46]. When these letters/characters are sequentially put together to form words the complexity increases due to the fact that some words might have their letters running together as shown in Figure 2.1.2.1 and Figure 2.1.2.2 below. These forms of writing require segmentation to facilitate the recognition of characters/letters.

Shape discrimination for similarly shaped characters is difficult for machine recognition. Some characters have similar shape such as *‘U’* and *‘V’*, *‘C’* and *‘L’*, *‘a’* and *‘d’*, and *‘n’* and *‘h’*. There are also difficult shape distinctions between certain characters and numbers such as *‘O’* and *‘0’*, *‘l’* and *‘1’*, *‘Z’* and *‘2’*, *‘S’* and *‘5’* and *‘G’* and *‘6’*. These characters can only be distinguished by context. On the other hand some of the upper case and lower case letters have similar shapes. Some of these letters among others are *‘C’* and *‘c’*, *‘K’* and *‘k’*, *‘O’* and *‘o’*, *‘P’* and *‘p’*, *‘S’* and *‘s’*, *‘T’* and *‘t’*, *‘X’* and *‘x’*, *‘Y’* and *‘y’* etc. The major distinction we can think of is probably their size and sometimes the position of the letters in reference to the baseline. These among others are the potential problems English Language character recognition might face.

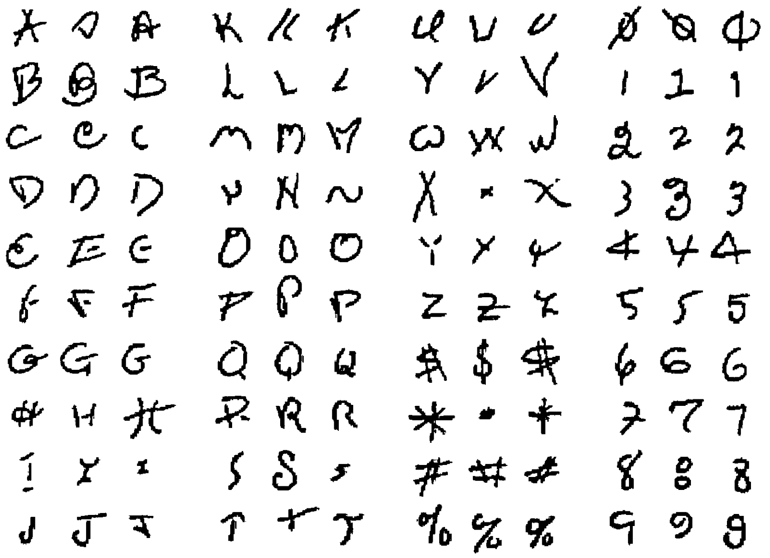
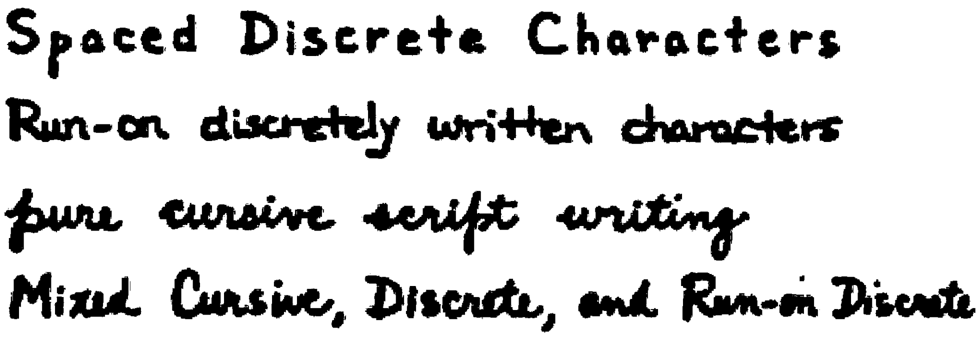


Figure 2.1.2.1 Examples of the different forms of handwritten symbols and characters.



Spaced Discrete Characters

Run-on discretely written characters

pure cursive script writing

Mixed Cursive, Discrete, and Run-on Discrete

Figure 2.1.2.2 Examples of different types of English Language writing

2.2 OVERVIEW OF PROCESSING REQUIRED FOR SIGNATURE-BASED ACCESS

The online handwriting signature recognition and verification system could be grouped under four major parts as illustrated in Figure 2.2.1 below. The main part for recognizing and verifying the handwritten signature include the data acquisition which is the real time handwritten signature collection using special hardware such as the three input devices. The second part for recognizing and verifying handwritten signature is preprocessing. This includes scaling, rotation, shifting, and filtering operations. There is also feature extraction which deals with the process of measuring the individual features of a signature. Feature matching and decision making which is achieved by comparing the reference signature with the extracted features and based on decision rules decision is made and the outcome determines whether the signature is genuine [56], [58].

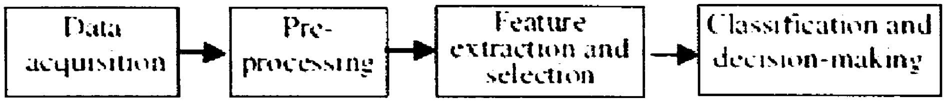

Data acquisition
Pre-processing
Feature extraction and selection
Classification and decision-making

Figure 2.2.1 online signature verification system

Alternatively, the design and implementation of an on-line dynamic signature verification system could involve data acquisition, feature extraction, feature selection, decision making and performance evaluation [26], [27], [28]. Another approach for capturing and processing of handwritten signature is categorized into six different stages and each stage is responsible of fulfilling the goals setup by the first module as listed below [28].

1) Acquisition stage: during this stage a handwritten signature is captured using a digitized hardware such as the Bluetooth enabled pen.
2) Preprocessing stage: the handwritten data is pre-segmented into strokes and filtered to eliminate the noise during the capturing of the signature.
3) Recursive sampling: the skeleton of the signature is generated by the use of recursive sampling of the resulting points by splines [16], [3].
4) Segmentation into stroke: the division of the written data into small groups called segments. The segmentation of the skeleton is done based on the curvature changes [29].

5) Windowing: windows are created around the stroke regions based on the outline of the stroke.

6) Learning stage: this consist of the adjustment of the size of each window and also selecting a prototype signature among the referenced data.

After going through these stages, the distance between two given pairs of signatures is calculated, this determines how similar the two signatures are from each other. That is the smaller the distance between the two signatures the greater the similarity and likewise the bigger the distance between the two signatures the smaller the similarity. For the purpose of this research, the online signature verification system as displayed in Figure 2.2.1 will be implemented.

There is a greater anticipation that most handwritten signature should be consistent in terms of the time, rate, force and shape during the writing of the signature and after the signature has been written. Typically, there is exhibition of similar temporal variations over the production of similar handwritten curves. In general, the speed along high-curvature curve segments is low and relative to the speed along low-curvature curve segments. The average overall speed vary greatly from one instance of a pattern to another irrespective of whether we are producing our own pattern or forging someone else's [30]. This observation suggests that at least the requirement of consistency over time during signature production is of limited value beyond that of consistency over shape. At any given rate, two signatures produced by the same individual irrespective of the velocities and forces used in generating the signature, it is of high necessity that the shapes of the signed signatures should match closely [30].

2.2.1. PREPROCESSING OF HANDWRITTEN SIGNATURE

As shown in Figure 2.2.1, Preprocessing is the first major step of processing and it involves the segmentation of the signature (Required for Offline signature capture), cleaning and smoothing the strokes. Some words are written together such that they form one long stroke. When there is a long stroke such as cursive writing, there is the need of isolation. The preprocessing where there is the need for isolation of various writing units prior to their recognition is termed as external segmentation and segmentation which does not require isolation before recognition is also termed as internal segmentation [46].

Noise is one of the factors to consider during the preprocessing of handwritten signatures. There are many techniques and algorithms that work well to reduce the noise during and after the capturing of the handwritten signature. The origination of noise could possibly be attributed to the hand motion of the author, inaccuracy of pen down indications, digitization process etc. Some of the techniques that could be used to reduce noise before and after handwritten signature acquisition are smoothing, filtering, wild point correction, dehooking, dot direction and stroke connection. Smoothing technique usually average the point with its neighbors, that is average a point with previous points permitting the computation to proceed as each point is received [51] [52]. Filtering is done to eliminate duplicate data points during the capturing of the data. Wild point correction is also done to eliminate the spurious points that may occur occasionally by the hardware used. Since acceleration of hand motion is limited by the forces of muscular contraction and the masses of hand and pen, the high acceleration or the velocities which is the changes in distance can help in wild point's detections [53]. Dehooking eliminates hooks that may occur both the beginning and at the end of each stroke. Hooks normally occurs because of inaccuracies of pen-down detections and too rapid motion in placing and lifting the pen. Dot correction reduces the dots to single point and the stroke connection eliminates extraneous pen lifts. That is it connects strokes that might have small distances between a pen up and subsequence pen down.

According to [31], [32], the temporal characteristics of the production of an on-line signature are key factors for signature verification. Verification of on-line signature relies upon either comparing features (Time, speed, acceleration, force pressure, etc.) of signatures or comparing temporal functions captured during signature production. There is a higher possibility of achieving better performance when both are implemented within a system. The approach that depends on comparing temporal functions performs better as compared to the approach that depends on comparing features alone [32]. The key ideas that underline the approach adapted by [30] are harmonic mean, jitter, aspect normalization, parameterization, and sliding computation window. With harmonic mean, two errors are combined and their root weighted mean square is computed. Based on the computation an ellipse is formed and generalizing the ellipse forms a super ellipse. Jitter normally occurs when an attempt is made to either make a copy or trace of an existing writing. There is constant correcting of the writing to conform to the original copy which results in the jitter exceeding the quantization error of the system.

Aspect normalization is most at times implemented because writers usually don't equally write their signature along the horizontal and vertical dimensions. The same writer might write their signature bigger and shorter and at different times and later times make their signature taller and longer. The one-to-one mapping of a subset of the original writing to the subset of the test data is referred to as parameterization. The original writing produces a parameter which makes locating of any point of the writing much easier if not simple. Once the mapping is done then functions of the subsets could be well described with their properties. After parameterization, the characteristic of the signature is derived from the center of the mass, torque and moments of inertia of the signature computed over a window which slides along the length of the signature. These thoughts among others could be put into three distinct component–normalization, description and comparison [30].

2.2.2 FEATURE EXTRACTION

Algorithm for signature verification process can basically be grouped into two kinds namely parameter and function methods [56]. Darwish and Auda in their signature verification research used neural networks as classifier when comparing the signature features [57]. They also made reference classifying most often used features in moment features and topological features. Moments and functions of moments were used as pattern feature and the computation of the moments require only one pass over the image. Topological features are best explained with examples such as shadow. Each pixel is projected onto the nearest vertical, horizontal or diagonal axis circles and sectors. The normalized image is divided into a number of concentric rings and a number of sectors. The mean distances where the normalized images are divided into a number of sectors, the quadrant feature where the normalized images are divided into 4 or 16 square regions and any number of features could be measured on each of the regions separately. Lam and McCormack used Fourier transform to verify signature [58], [59].

Mahakrishnan and Paulik proposed a different direction of signature verification by representing signature by a jump non-stationary autoregressive model [60], which treats the signature as an ordering of unique curve type and each of the curve type is represented by an autoregressive model. The verification process they adapted was similar to the procedure used for speaker verification. With this verification process the unknown author writes a sample signature and in addition to the sample the author provides an identity claim from amongst the writer population. The distance between the two is computed from a selected metric and they are compared. If distance between the claimed reference writer and the test writer as computed is less than the selected threshold, the author's claim is accepted and if otherwise the claim is rejected.

2.2.2.1 FRACTAL FEATURES

Fractal theory which is another algorithm for on-line signature recognition has been successfully applied to computer graphics, image compression and different fields of pattern recognition. Fractal theory of iterated function systems which has extensively been investigated in computer graphics and image compression [34, 35], has a potential in different fields of pattern recognition such as face recognition [36, 37, 38], character and digit recognition [39, 40, 41, 42], signature verification [43] and texture recognition [44]. The fundamental principle of fractal coding consists of the representation of any image by a contractive transform of which the fixed point is too close to the original image. The procedure for finding a fractal model for a given image is called encoding, compression, or searching for a fractal image representation [45]. Many researchers have implemented the fractal codes obtained during the encoding process in different classifications [36], [37], [41], [42]. The properties of a fractal theory based on the fixed point theorem of Iterated Function Systems have also been exploited by some researchers, called Fractal Transformation. According to [43], the distortion between an input pattern and the pattern after decoding is the basic idea for classification.

Online signature recognition system deals with a time ordered sequence of points based on the pen positions [45]. The number of points in a signature locus depends on the sampling rate of the tablet digitizer and also on the speed of writing, so there is a need for a preprocessing step to smoothing and resample the signature into a number of spatially uniform sample points [45] as shown in the Figure 2.2.2.1.1 below. This will not be implemented in this research work due to the lost of velocity information after smoothing and resample, which is needed for the purpose of this research. The partitioned ranges of the signature are then mapped onto their respective domains according to a given algorithm. To find the best match for the range of segments, each transformed segment is resample into a number of different points. The centroid of the segment is determined based on the factors and parameters under consideration. The most similar segments from the list of all transformed segments are chosen as the corresponding domain segment.

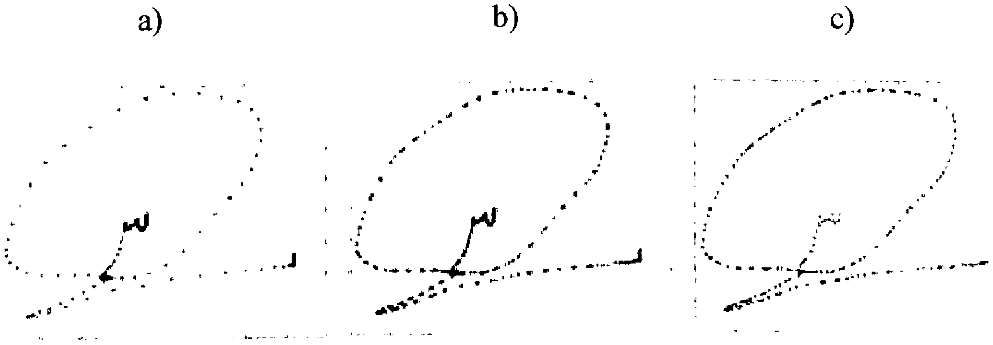


Figure 2.2.2.1.1 Fractal preprocessing steps a) Original locus b) interpolating curve c) spatially uniformly resample locus.

2.2.2.2 FOURIER DESCRIPTORS

The Fourier descriptors method was first introduced by Zahn and Roskies in the early 1970s. This method describes the shape in terms of its spatial frequency content. Fourier descriptor method mainly consists of computation of boundary pixels, use of shape signature function, and computation of Fourier descriptor [70]. When the boundary pixels are computed, a pixel set can be formed. Shape signature functions are used to compute shape signatures from the boundary pixels set. For the online signature recognition, the actual x-y coordinate pairs can directly serve as the inputs to the Fourier Descriptor calculations. Complex coordinates, curvature function, cumulative angular function, and centriod distance are the commonly used shape signature functions. Fourier descriptor methods using these shape signature functions are compared in [69]. As shown by Zhang and Lu [69], Fourier descriptor method using centriod distance outperforms Fourier descriptor methods using other shape signature functions in terms of overall performance. Fourier descriptors are well known for capturing boundary information and it is invariant to translation, rotation and scaling. Fourier descriptors uses contour information and signatures that are mostly used are the complex coordinates; centroid distance, curvature signature and cumulative angular function. Fourier descriptors are also widely used to represent closed planar contours for the purposes of determining the attributes of shapes [71}.

According to Lin and Chellappa [1], the method used to acquire the estimates for the Fourier descriptors minimizes the sum of the least square fit of the data subject to the condition that the number of the missing boundary points and the perimeter2/area of the shape are not known. Elliptic Fourier descriptors have promising properties in statistical classification schemes for single and vectored handwritten symbols [6]. The paper projected that both SysScan's Georec system and Intergraph's data capture systems implemented the vector representation for object recognition. The automatic recognition of both the maps and the drawings were all single handwritten symbol recognition that requires the best feature measurement method. Fourier descriptors offer a lot of advantages. They allow a reconstruction of a symbol based on the descriptors alone. This property makes it possible to extract all relevant structural information from a symbol, if a sufficient number of descriptors are included in the feature vector.

The elliptic Fourier descriptors of one symbol class usually give rise to a unimodal distribution in feature space. This is well modeled by a multivariate normal distribution [6]. If the symbols are rotated only to a limited extent from the vertical orientation, the rotation invariant elliptic Fourier descriptors frequently allow a discrimination even of symbol classes which differ only in their rotation angle. A disadvantage of the elliptic Fourier descriptors derived from vectorized symbols is the need for subclasses in the statistical classification scheme. For each symbol class the size of the training set has to be increased in proportion to the actual number of subclasses to get good statistical density estimates. One major drawback of the Fourier descriptor is its inability to differentiate symbols which differ only by its rotation angle [7]. A typical illustration of this drawback is the symbol '66' and the symbol '99'.

2.2.3 CLASSIFICATION AND DECISION MAKING

There are several recognition methods in use today depending on the various distance functions [14]. According to [19], [20], [21], the distance between the test sample and the training templates are measured as a simple distance or as a Euclidean distance. This section will discuss feature-based recognition methods and String-based recognition method. Another taxonomy within which we can classify signature recognition methods is into statistical or structural approach which has been discussed in the survey [66].

2.2.3.1 FEATURE-BASED RECOGNITION METHODS

A wide variety of classification techniques have been proposed and most of these classifications are based on Bayesian decision rules. This aim to minimize the classification error or a generalized risk function provided the probability density function of each class is known. The Bayesian classification is realized by parametric or non-parametric techniques. In parametric classification, the probability distribution is often assumed to have a Gaussian form which end up either being a quadratic classifier or a linear classifier. In non-parametric classification, either the conditional probability or the posteriori probability is estimated directly from the training samples. The k-nearest-neighbor is one of the most popular non-parametric classifications whereby the probabilities are estimated from the frequency of nearest neighbors to the unknown pattern. The performance of k-nearest neighbor classifier asymptotically approximates the Bayesian classifier if the number of training samples approaches infinity [65].

Amongst the different types of classifiers described by Anoop [3], the k-Nearest Neighbor (kNN) classifier is the most efficient, simple and effective nonparametric classification method. This method gives a high recognition rate and allows efficient implementation [63]. This method is a simple but powerful classification technique [64]. The original kNN algorithm was put forward by T. M. Cover and P. E. Hart as discussed in [78]. With this approach, each training sample will be used as a prototype and the corresponding test sample will be assigned based on the closeness of the prototype. The error rates for different classifiers were experimentally described by Anoop [3] as shown in the Table 2.2.3.1.1 below. From the figure, we could deduce that the 11-nearest neighbor that was normalized had a percentage error of 15.2 and that of 15-nearest neighbors was 15.4. This table indicates k-nearest neighbors with their features normalized, producing the best performance since it is one of the simplest algorithms.

The KNN algorithm has simple implementations, it is analytically tractable, and is nearly optimal in the large sample limit [67]. The main disadvantage of this algorithm is that its non-parametric algorithm's need to consult a reference sample during each classification. Another obvious problem with kNN algorithm is that, when the density of the training data is uneven it may decrease the precision of the classification if the first k nearest neighbors is considered and the difference of the distances are not considered. To solve this problem, a fuzzy sets theory could be used by constructing a new membership function based on document's similarities [77]. With the improvement of kNN using the fuzzy set theory, questions such as how to improve decision rules, how to select k, how to select the feature set to make the classification result better and their effect to each other in classification performance will be addressed. For readers who are interested in the fuzzy kNN algorithm are recommended to read [77].

CLASSIFIERS	REMARKS	ERROR RATES (%)
Nearest Neighbor	**No Normalization**	**35.8**
Nearest Neighbor	**Normalized Features**	**17.6**
5-Nearest Neighbor	**Normalized Features**	**15.4**
11-Nearest Neighbor	**Normalized Features**	**15.2**
Bayes Quadratic	**Gaussian with Full Covariance**	**25.5**
Mixture of Gaussian Distribution	**Diagonal Covariance**	**25.5**
Decision Tree	**C5.0**	**16.1**
Neural Network	**One hidden layer with 25 Nodes**	**14.3**
SVM	**RBF Kernel**	**13.5**

Table 2.2.3.1.1 error rates for different classifiers

2.2.3.1.1 NEAREST NEIGHBOR ALGORITHM

The Nearest Neighbor Algorithm is a method of classifying objects based on the closest training sets. Given an unknown feature vector *'x'* and a distance measure then out of the 'N' training vectors you identify the *'k'* Nearest Neighbor. *'k'* is chosen to be odd for a two class problem and in general not to be a multiple of the number of classes 'M'. Out of the *'k'* samples, you identify the number of vectors *'k_i'* that belong to class w_i, i=1,2,...,M. Obviously Σk_i=k. Then you assign *'x'* to the class *'w_i'* with the maximum number 'ki' of samples. Various distance measures can be used including Euclidean and Mahalanobis distance. The simplest version of the algorithm is for k=1, known as the Nearest Neighbor rule. In other words, a feature vector *'x'* is assigned to the class of its nearest neighbor provided that the number of training samples is large enough; this simple rule exhibits good performance [80].

It was shown by Duda [7] that, as $N \rightarrow \infty$, the classification error probability for the Nearest Neighbor classifier is at most twice that of the optimal classifier. The asymptotic performance of the *'k'* Nearest Neighbor is better than that of the Nearest Neighbor. As $N \rightarrow \infty$ the performance of the *'k'* Nearest Neighbor tends to be the optimal one. It is only in the limit as 'N' goes to infinity that we can be assured of the nearly optimal behavior of the 'k' Nearest Neighbor rule. For large 'N' and small Bayesian errors, there is a greater expectation that k=3 Nearest Neighbor classifier will give performance almost identical to that of the Bayesian classifier.

Lets assume that the error probability of the Bayesian classifier is of order 1%, then the error resulting from a k=3 Nearest Neighbor classifier will be of the order 1.03% and this approximations improve with higher values of *'k'*. Under the assumption of large 'N', the radius of the hyper sphere (Euclidean Distance) centered at *'x'* and containing its *'k'* Nearest Neighbors tends to zero [81]. That is at very large 'N', we expect the space to be densely filled with samples. Thus, the *'k'* neighbors of *'x'* will be located very close to it and the conditional class probabilities at all points inside the hyper sphere around *'x'*, will be approximately equal to $P(w_i/x)$.

Furthermore, for large *'k'*, majority of the points in the region will belong to the class corresponding to the maximum conditional probability. Thus the 'k' Nearest Neighbor rule converges to the Bayesian classifier. However, in conclusion, it can be stated that the Nearest Neighbor techniques are among the serious candidates to be adopted as classifiers in a number of applications [80].

2.2.3.2 STRING-BASED RECOGNITION METHODS

In this section, Dynamic Programming and Dynamic Time Warping (DTW) are described as two the different methods for string recognition. These methods recognizes handwritten signature by looking at the signature as a string instead of individual characters or symbols. A string comprises of a sequence of characters which has been classified into a defined group/classes. The major disadvantage that comes with the implementation of string-based recognition is that the size of the pattern will become large but viewing the signature as a whole string/word avoids segmentation.

2.2.3.2.1 DYNAMIC PROGRAMMING

Dynamic Programming (DP) matching method is a well-known and an effective method for on-line handwritten character recognition. This method is sometimes referred to as the forward-backward and when working with probabilities it is referred to as Viterbi algorithm [9]. Dynamic programming addresses the issue of restricted memory search in problems composed of multiple interactions and interrelated sub-problems. Dynamic programming method has three possible costs for its current state. If a character is shifted along in the shorter string for better possible alignment, the cost is *'1'* which reflects in the column score. If a new character is inserted the cost is *'1'* which reflects in the row score. If the characters to be aligned are different, you shift and insert resulting in a cost of *'2'* (*'1'* for shift and the other *'1'* for insert). If they are identical the cost is *'0'* which reflects in the diagonal. A minimum edit difference between the two strings (Lavenshtein distance) could be specified as the number of character insertions, deletions and replacements necessary to turn the first string (source) into the second string (target).

2.2.3.2.2 Dynamic Time Warping (DTW)

Dynamic Time Warping algorithm was originated from the field of speech recognition where it is a key component of speaker specific isolated word recognizer [73]. The DTW is a formulation to find a warping function that provides the least distortion between any two given patterns. The optimum solution is determined through the dynamic programming methodology. DTW can be viewed as a pattern dissimilarity measure with embedded time normalization and alignment. The algorithm could be extended to multiple patterns greater than two resulting in the multi-pattern dynamic time warping (MPDTW). MPDTW can be used to determine the optimum path in the multi-dimensional discrete space to optimally warp all the number of patterns jointly [74]. DTW-algorithm has been successfully introduced in online signature verification. Dynamic time warping algorithm is implemented on online signature verification by extracting stable and idiosyncratic features out of the way user's signs. The pen-tip position, the forces exerted on the surface by the pen and signals are used by DWT-algorithm in verification process.

According to Ronny and Luc [73], there isn't any adaptation when migrating from speech recognition to signature verification. As mentioned in [74], the problem for which the algorithm is intended should have the following properties;

1) The pattern to be compared is time-sampled with a common and constant sampling period.
2) There is no prior knowledge about the relative importance of different parts of the patterns.

Condition '1' can generally be satisfied easily but condition 2 will be a challenge if your research is exposed to a large set of dataset. A complete description of this algorithm and its implementation can be found in [73]. Interested readers should read further [73] and [74]. DTW has been successfully used in many domains but the crucial observation is that the algorithm may try to explain variability in the *Y-axis* by warping the *X-axis*. This can lead to unintuitive alignments where a single point on one time series maps onto a large subsection of another time series [75]. DTW algorithm is well illustrated and documented in [75] [76].

2.2.3.3 REVIEW OF FUZZY LOGIC

The structure of fuzzy logic gives a unique representation of natural methods in support of human decisions and reasoning. Basically, fuzzy logic is a precise logic of imprecision and approximate reasoning. More specifically, fuzzy logic may be viewed as an attempt at formalization/mechanization of two remarkable human capabilities. The first of these is the capability to converse, reason and make rational decisions in an environment of imprecision, uncertainty, incompleteness of information, conflicting information, and partiality of truth in an environment of imperfect information. The second is the capability to perform a wide variety of physical and mental tasks without any measurements and any computations [13]. Fuzzy logic is much more than a logical system. It has many facets, including: logical, fuzzy-set-theoretic, epistemic and relational. Most of the practical applications of fuzzy logic are associated with its relational facet [13].

Fuzzy logic has been instrumental, cutting across all sectors of research, ranging from information systems, decision processes, the medical field, the engineering field, the energy sector, health sector and mechanical sectors. This paper will be focusing on implementing fuzzy logic in the information technology sector, specifically pattern matching.

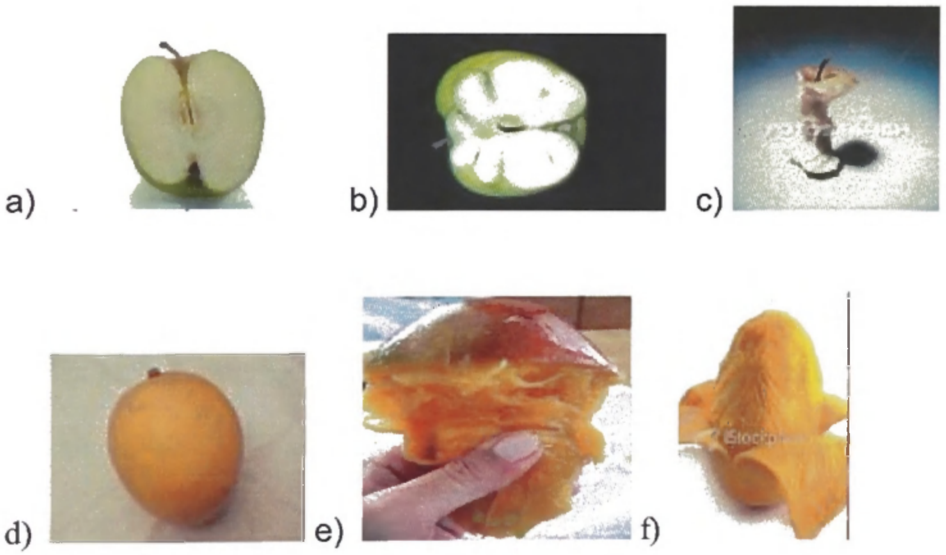
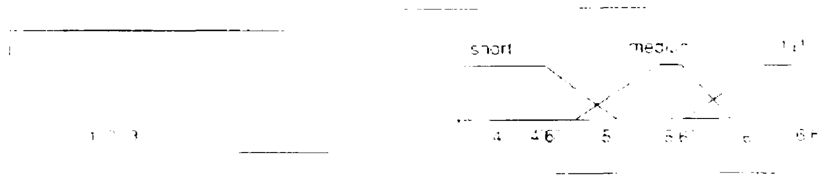


Figure 2.2.3.3.1 Mongo fruit and an apple fruit a) cross-section of the apple displaying the different sections b) partially eaten apple to display the core of the apply c) fully eaten apple showing the uncertain part of the fruit left on the core d) fully ripe mango fruit e) partially eaten mango displaying the seed f) fully pealed mango showing the certainty of the back and the fruit

To understand the concept behind fuzzy logic, let us relate this concept with two different fruits, the apple and mango as shown in Figure 2.2.3.3.1 above. We can decompose a mango fruit into three different discrete sets, the outer cover of the mango fruit, the fruit itself and the seed which forms the core of the fruit. The decomposition of the apple fruit into two discrete sets become difficult if not impossible because at what point should we stop biting the apple fruit to separate the fruit from the core. There is a certain level of uncertainty when differentiating between the two sets. The law of excluded middle states that an element cannot belong to both a set and also to its complement. For any given set, an element belongs to either the set or the complement of the set. This concept of the excluded middle is a critical foundational concept for traditional, i.e. Bayesian, probability, and its refutation is likewise the key to fuzzy logic. Going back to our fruit example, the mango will have no problem following the law but with the apple we will find problem. The area between the apple fruit and the core of the apple is not well defined. Technically we can redefine this fruit using the fuzzy sets.

Fuzzy set theory gives room for its members to have degrees of membership [9]. Luger in [9] stated that, "Zedeh's theory expresses lack of precision quantitatively by introducing a set membership function that can take on values between *'0'* and *'1'*". All elements that belong to the apple fruit is given a discrete value of *'1'*, all elements that belong to the core of the apple is given the value *'0'* and all other elements that have characteristic features of both sets would have a value between *'0'* and *'1'*. These values allocated to the respective sets are the degree of membership since all might not be discrete.

A standard example of a fuzzy set as displayed in Figure 2.2.3.3.2, could be drawn from a set *'S'* and a member of the set *'s'* while a fuzzy subset *'F'* of *'S'* is defined by a membership function *"mF(s)"* which measures the degree to which *'s'* belongs to *'F'*. Let *'S'* be the set of positive integers and *'F'* a fuzzy subset of *'S'* (small integers), various integer values can have a possibility distribution defining their respective fuzzy membership in the set of small integers as *mF(1)=1.0, mF(2)=1.0, mF(3)=0.9, mF(40)=0.8 ... mF(50)=0,001* etc For the fact that the positive integer is a small integer, the membership function creates a possibility distribution across all the positive integers (S). Fuzzy set theory is more concerned with the rules for computing the combined possibilities over expressions that contain fuzzy variables instead of how possibility distributions are created [9]. For the fuzzy set representation of the set of small integers as shown in Figure 2.2.3.3.2a below, each integer belongs to a set with an associated confidence measure. In other words each element which is in a set must have a membership or confidence associated with it with a value between *'1'* and *'0'*. Let's consider height as an illustration of this principle. Height of people could be relative depending on your reference point. In our example we will classify height into short, medium and tall. Each classification will represent one membership function as displayed in Figure 2.2.3.3.2b below. There is a possibility of one person belonging to more that one membership function, for example a 5 ft and 9inches male belongs to both the set of medium as well as the set of tall males.



a) **Small integers** b) **Short, medium and tall males**

Figure 2.2.3.3.2 Fuzzy set representation

One of the major factors to consider when working with fuzzy logic is determining the membership functions. According to [8], the major difficulty in determining the membership functions is the use of linguistic labels and descriptors on variables because of the contextual dependencies of the linguistic descriptors. That is what might be considered as a smooth surface in one field might be considered as rough in another field of measurement. Linguistic descriptors are relative based on the context been implemented. This requires knowledge about the descriptor, the process of the operation of the descriptor as well as the control procedures of the descriptors. The knowledge about the operation is based on rules. The combination of the rules and the membership function enhances and guides the decision making process.

Fuzzy logic is an operator which consists of three sub-operations namely the fuzzifier, rule evaluator and defuzzifier. The input to the fuzzy operator is defined according to a range of values which is mapped to a set of attributes, namely the fuzzy set. The fuzzy set consists of elements which in this case will be called linguistic descriptors such as high, low, medium etc. Unfortunately, the measuring devices do not provide fuzzy membership values, but rather provide actual values (crisp). Therefore the first step in a fuzzy logic system is to convert the crisp measurements into fuzzy membership values which are accomplished by the fuzzifier. The assigned value is calculated using the membership function. After several processes the output of the fuzzifier becomes an input to the rule evaluator. The rule evaluator evaluates and calculates the strength of the fuzzy input and maps them to the defuzzifier which determines the value for the fuzzy data as shown in Figure 2.2.3.3.3 below.

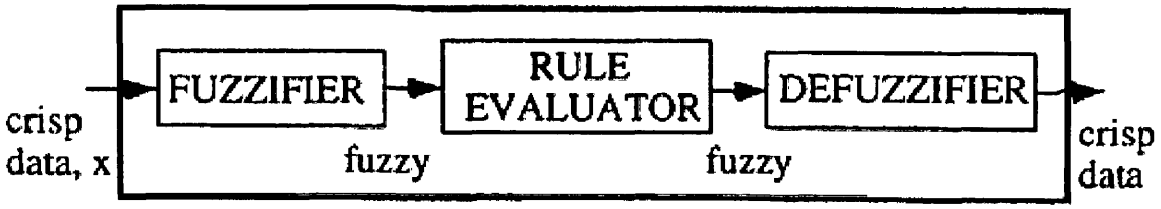


Figure 2.2.3.3.3 functions of the fuzzy logic operator

Ramot, Friedman, Langholz and Kandel stated in their paper that, in complex fuzzy logic, inference rules are constructed and fired in a manner that is closely parallel to a traditional fuzzy set [10]. The main idea is that, the sets used in the reasoning processes are complex fuzzy sets which are characterized by complex–valued membership function. The range of complex-valued membership function is derived from the traditional membership function which is *'0'* and *'1'*. The method for deriving a membership set in terms of complex numbers. Complex fuzzy set theory allows a natural extension of fuzzy logic to problems that are either very difficult or impossible to address with one–dimensional grade of membership [10]. During the derivation of the complex fuzzy set, several set theoretic operation should be performed. Among these set theoretic operations are complex fuzzy union, complex fuzzy intersections and set aggregation which was termed as vector aggregation. Complex fuzzy set is well defined, explained and examples are given to illustrate the idea. Interested readers who are looking to solve more complex problems should consider reading this paper [10].

Fuzzy logic could be used in two different senses. Fuzzy logic is a logical system which is an extension of multi-valued logic and is intended to serve as logic of approximate reasoning. On the other hand fuzzy logic is synonymous with the theory of fuzzy sets, which is theory of classes with unsharp boundaries [11]. What is gained through fuzzification is greater generality, higher expressive power, an enhanced ability to model real-world phenomena and most importantly, a methodology for exploiting the tolerance for imprecision that is a methodology which serves to achieve tractability, robustness and lower solution cost [11].

In general, knowledge is encoded initially in either simple or complex linguistic expressions of a natural language. These linguistic expressions are first transformed into more understanding expressions (axiomatic expressions) known as propositions and predicates using the principles of set theories, membership functions and their respective connectives. It is further transformed into computational expressions with the assignment of numbers to the symbols which was determined during the first transformation [12].

Naturally, there is a restricted version of native human knowledge and intelligence encoded in our biological neuronal constructs, but it appears that most people in everyday life reason in somewhat a similar manner where information granules are identified and processed with linguistic terms of a natural language via human information processing capability [12]. Within the scope of scientific abstraction, there is a generation of short hand notations to represent linguistic variables. Variables such as inventory, demand, and production, are given symbols *'X', 'Y', 'Z'*, respectively, and linguistic values such as “low”, “medium” and “high” are also represented with fuzzy set symbols such as *'A', 'B'*, and *'C'*, respectively [12].

2.2.3.3.1 Fuzzy Classifiers

One might ask what a fuzzy classifier is. According to Ludmila [84], there is no clear-cut definition of fuzzy classifier. Instead, Ludmila tried to define fuzzy classifier using these three illustrations. If we represent 'x' as a vector in an n-dimensional real space (Rn) and 'w'= {w1, w2, w3…wn} represent a set of class labels then a classifier is any mapping of (D: Rn->W). With this mapping in mind the first definition of fuzzy classifier is any classifier which uses fuzzy sets either during its training or during its operation [84]. The second definition of fuzzy classifier is any possibilistic classifier for which $\sum \mu i(x) = 1$ from i=1 to 'c' [84]. The third definition of fuzzy classifier is a fuzzy if-then inference system (that is fuzzy rule-base system) which yields a class label for x [84].

The three definitions are embedded in each other somehow. This conclusion was drawn from the fact that, the third definition is based on fuzzy set and since definition one uses fuzzy set we can clearly say definition three lies in definition one. Classifiers that use fuzzy sets example fuzzy k-nearest neighbor methods, do not necessary produce class labels that sum up to one nor are they rule-based [84]. Therefore there are some areas that definition one will cover but definition two will not account for.

2.2.3.3.2 Why Fuzzy Classifiers?

This question was best answered by Ludmila [84] where five points were made clear about why we should use fuzzy classifier. These five points are listed below.

1) In some problems, there is insufficient information to properly implement classical pattern recognition methods.

2) Users often times need additional information such as the severity of the problem under study and not only the class label.
3) Often times the characteristic of the object or the class labels are conveniently represented in terms of fuzzy sets.
4) Expert's opinion about classification decision; features and objects are well processed by the mathematical tools which fuzzy set theory provides.
5) Fuzzy classifiers based on IF-THEN rules might be "transparent" or "interpretable".

Although there are reasonable ideas why fuzzy classifiers are important and easily implemented, there are some obstructions. Fuzzy classifiers become difficult to design if the classifier is based entirely on the expert's opinion. This is normally referred to as "knowledge acquisition bottleneck" [84]. Fuzzy classifiers do not offer an easy way of handling complex dependencies between the features. In order to ensure some level of transparency there is a need for linguistic reasoning which granulate the feature space [84]. According to Tickle, transparency is necessary only when dealing with small number of features and small number of linguistic labels defined on the feature. In problems of higher dimensionality, interpretation might not be feasible [86]. Since there is no rigorous theory, there is no theoretical methodology to design a fuzzy classifier for every instance [84].

Selecting a classifier for the problem under study could be a little bit difficult. But we should bear in mind that there is no such thing as the best classifier [84]. Classifiers applied to different problems and trained by different algorithms perform differently [87, 88]. Duin [88] said that, the performance of a classifier depends on the expertise and the willingness of the designer. The asymptotic behavior of some classifiers is known but these behaviors do not guarantee good performance. There have been a lot of experimental studies finding other classifiers to be better than others, but studies are based on extensive experimental evidence using a number of simulated and real data set [84]. It becomes hard when judging who is right about their experimental studies and who isn't. Some of the things to consider when selecting your classifier are the error rate, experimental design and classifier complexity.

2.3 EVALUATING SIGNATURE VERIFICATION SYSTEMS

For a signature verification system to be very useful, the system must commit few errors in practice [30]. Most times the best way to avoid introducing a system into the market place with errors is to perform field test. Many organizations and firms ignore field test due to the cost and time needed to go through this practice. There are two criteria that could be used to evaluate signature verification system [30]. The first criteria state that, whenever you try the system it must work. That is, signature verification system should be able to recognize similar scribbles consistently, must detect when there is forging of someone else's signature and must deny scribbles that are visually desperate from the original. The second criteria state that when you test the system with large databases it must exhibit low statistical error rates. That is determining the percentage of false accepts as a function of the percentage of false rejects.

It is apparent that two types of errors can result from a verification test - false acceptance (FA) of a fraudulent claim and false rejection (FR) of a genuine claim. In addition, it is clear that the choice of the threshold will determine the relative occurrences of false acceptance and false rejection types of errors. A reduction of the threshold will decrease the incidence of false acceptance errors while increasing the incidence of false rejection errors. An increase in the threshold will do exactly the opposite. In practice, a technique that is often used as a figure of merit of a verification system is to find a threshold that equalizes the probabilities of false accept and false rejection. This involves the following sequence of operations. Intra-writer and inter-writer distances are generated between reference and test writers over the database using the chosen distance metric. Cumulative distribution functions are then plotted for the two sets of distances as a function of distance threshold. To clarify, these two distribution functions would then have percentage of intra-writer distances greater than the threshold and percentage of inter-writer distances smaller than the threshold as ordinates. The intersection of the two curves then provides the equal-error estimate and the corresponding distance threshold. It must be emphasized that having established a figure of merit for a verification system in this manner, one is not constrained to deploy the system with this threshold. Depending on the application involved, one could choose to bias the system favorably towards either false accept or false reject [27].

3. DETAILS OF RESEARCH

3.1 OVERVIEW OF TECHNICAL DETAILS

From our previous discussions, online signature recognition was compared to the offline signature recognition. For the purpose of this research work the online signature recognition will be implemented for the capturing of the handwritten signatures using three selected input devices. The three selected input device are the mouse, the I-Pen and the IOGear. At the end of this research work a conclusion will be drawn as to which input device will be most efficient and effective for file encryption and computer access based on the sensitivity of the device and how easily users adapt to the use of the device. Unlike other forms of online document which may be represented in different languages and scripts, this online signature study will be handwritten with the English language in mind.

In the signature recognition process, the main focus is to make as few errors as possible in the classification and decision making processes. There exist several approaches for improving the accuracy of a recognition system. To make few errors in the classification and decision making process, this research paper will be considering feature-based recognition method and fuzzy logic recognition method. Similarly, the k nearest neighbor (kNN) algorithm will be used as the baseline for the fuzzy k nearest neighbor (Fuzzy kNN). Finally, the fuzzy If-Then algorithm will be used in making the decision on either to accept or reject signatures.

There are six processing stages in this signature classification as presented in Figure 3.1.1. The steps involve the pre-processing of the signature, normalization, generating the membership functions, threshold and finally classify the signature.

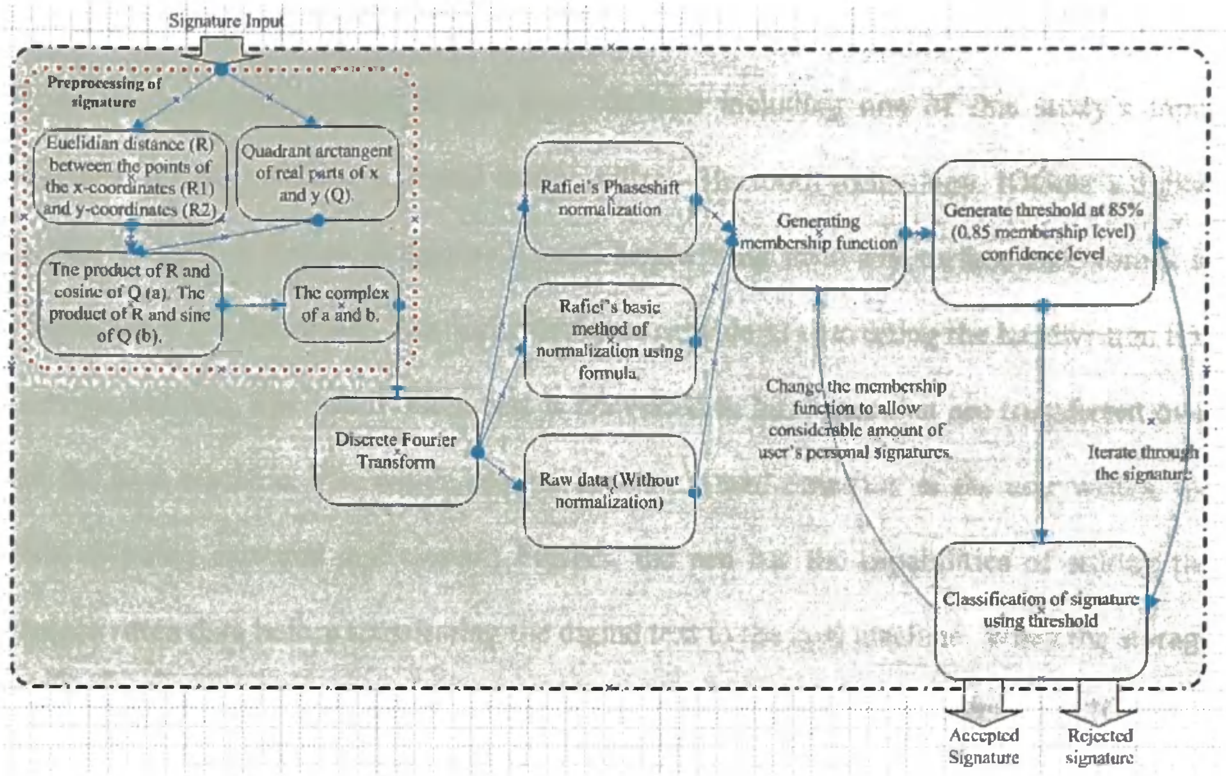

Signature Input
Preprocessing of signature
Euclidian distance (R) between the points of the x-coordinates (R1) and y-coordinates (R2)
Quadrant arctangent of real parts of x and y (Q)
The product of R and cosine of Q (a). The product of R and sine of Q (b).
The complex of a and b.
Discrete Fourier Transform
Rafiei's Phaseshift normalization
Rafiei's basic method of normalization using formula
Raw data (Without normalization)
Generating membership function
Generate threshold at 85% (0.85 membership level) confidence level
Change the membership function to allow considerable amount of user's personal signatures.
Iterate through the signature
Classification of signature using threshold
Accepted Signature
Rejected signature

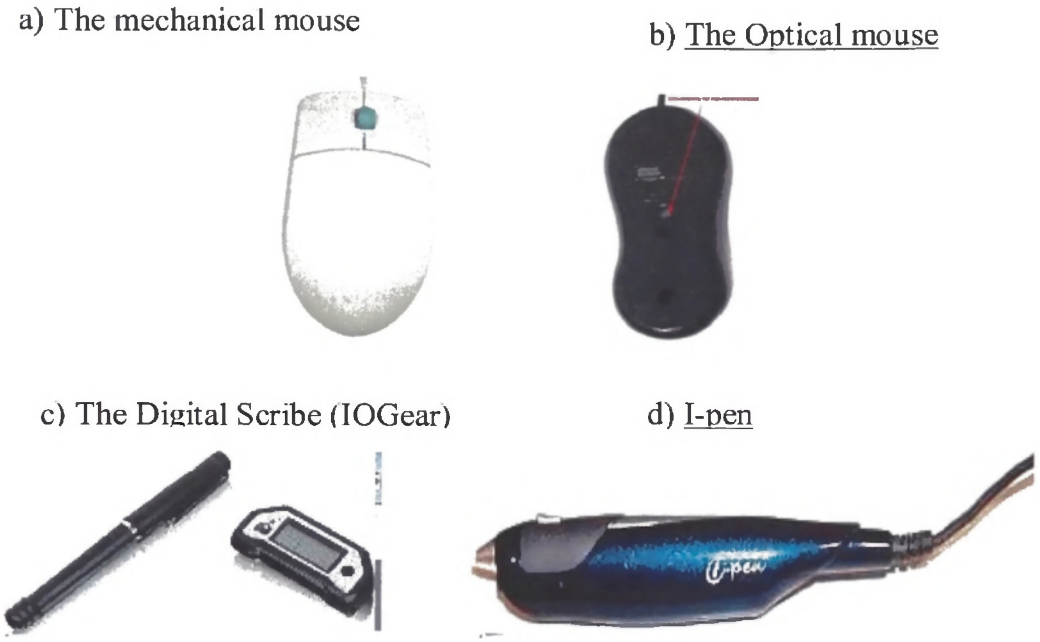
Figure 3.1.1 the processing stages of signature classification.

3.1.1 INPUT DEVICES

A computer mouse is computer input device used to control the cursor on the screen. The mouse is considered an input device because of its pervasiveness of use. The computer typically uses the mouse's *'X'* and *'Y'* position signals to manipulate the display of the computer screen allowing a user to control a program. The scope of this research will be limited to more commonly used and readily available wired mice, namely the electromechanical and optical mouse with a wired connection to the computer.

Conversely, a Bluetooth-enabled pen uses a technology widely known as Bluetooth. Bluetooth is a short range wireless technology used to create Personal Area Network (PAN) among nearby devices. This technology has been implemented since its invention to improve on the communication between nearby devices. This technology has been implemented in most of the input devices including one of this study's input devices, IOGear. IOGear's Digital scribe utilizes Bluetooth technology. IOGear's digital scribe is the first device to capture natural handwriting from any surface and store it in the receiver for future use. This input device is capable of converting the handwritten text into digital text. The handwritten text is converted to bits/bytes that are transferred over to the host machine. The whole process is done in real time (i.e. as the user writes, the text is being captured and stored.) Further, the pen has the capabilities of storing the handwritten data before transferring the digital text to the host machine. When the storage process is skipped and text is transferred directly to the host machine, the IOGear's digital scribe works like the wireless tablet. The IOGear's digital scribe uses an infrared sensor and ultrasonic transmitter in its base to detect hand movements and digitally record them [5]. Figure 3.1.1.1c is an example of the IOGear's digital scribe input device used in this study to capture participant's hand written signature.

Similarly, the I-pen is another version of a pen. I-pen is a digital pen that works like a mouse but has the capability for the user to write in their own handwriting. The I-Pen connects to the host using a universal serial bus (USB), which is also plug-and-play. This device intelligently recognizes handwritten text, converts the handwritten text to digital text and then transfers it to the host machine. I-pen, like the computer mouse, has a right click, left click and center scroll buttons. Further, the I-pen functions as a left click when the tip is pressed and it works as a right click when the button on the pen is pressed. Thus the I-pen was designed to work in two forms, a pen and a mouse alternating between these two different modes of operation. Figure 3.1.1.1d displays the I-Pen implemented in the study to capture the user's handwritten signatures.



a) The mechanical mouse

b) The Optical mouse

c) The Digital Scribe (IOGear)

d) I-pen

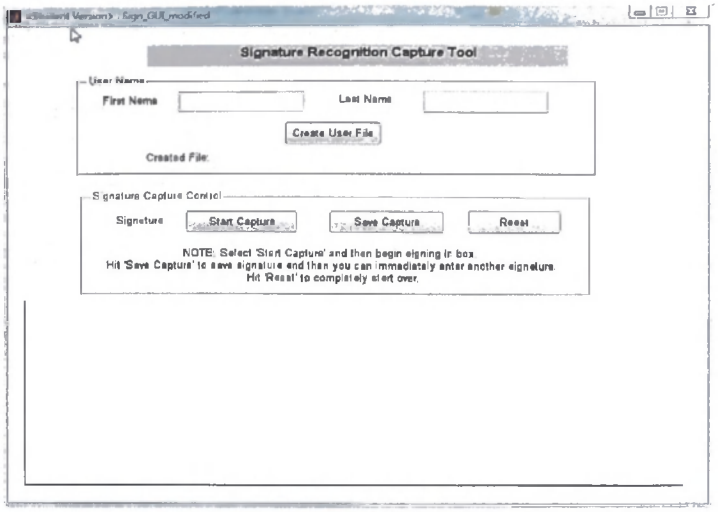
Fig. 3.1.1.1 the three input devices a) A mechanical mouse b) Optical mouse c) Digital Scribe (IOGear) d) I-pen

3.1.2 SIGNATURE CAPTURE

The technique and the technology used to capture online signature is one of the crucial factors in determining the accuracy and quality of the data captured for processing. Depending on the technology used, the capturing device can provide the *'x'* and *'y'* coordinate, Boolean representation of contact with the *'x'* and *'y'* plane, pressure, the angel at which there is a contact (tilt), the speed and how far away is it from the *'x'* and *'y'* plane. For the purpose of this research work, the selected capturing devices were designed to provide the *'x'* and *'y'* coordinate of the signature which determines the location of the input device during the capturing and the position of the signature when captured. The capturing tool was also designed to capture the speed at which users sign their respective names. The purpose of selecting these characteristic features was because the selected input devices which are the mouse, the I-Pen and the IOGear captures neither the pressure nor the angle at which the device is tilted. These input devices are readily available and less expensive.

Figure 3.1.2.1 displays the capturing tool used in capturing the handwritten signatures of users. The tool request of the user to enter his/her first name and last name to create user's file. After the user has entered the information needed, "Create User File" button is clicked to create the file. Once the file is created the user can go ahead and click on the "Start Capture" button to start signing their names within the space provided. When users make mistakes during the signing of their name, the reset button could be used to start over again. When the user is satisfied with the signature signed the "Save Capture" button is clicked to enable the user to save their signature.

During the course of this research, there was an anticipation of getting about 50 people to sign their names multiple times with the selected input devices. After the capturing process there were about 57 people who signed their names ten (ten) times for each selected input device. Figure 3.1.2.2a, Figure 3.1.2.2b and Figure 3.1.2.2c show examples of user's signatures using the mouse, I-Pen and the IOGear respectively.


<Student Version> : Sign_GUI_modified
Signature Recognition Capture Tool
User Name
First Name
Last Name
Create User File
Created File:
Signature Capture Control
Signature
Start Capture
Save Capture
Reset

NOTE: Select 'Start Capture' and then begin signing in box.
Hit 'Save Capture' to save signature and then you can immediately enter another signature.
Hit 'Reset' to completely start over.

Figure 3.1.2.1 Snapshot of the capturing tool

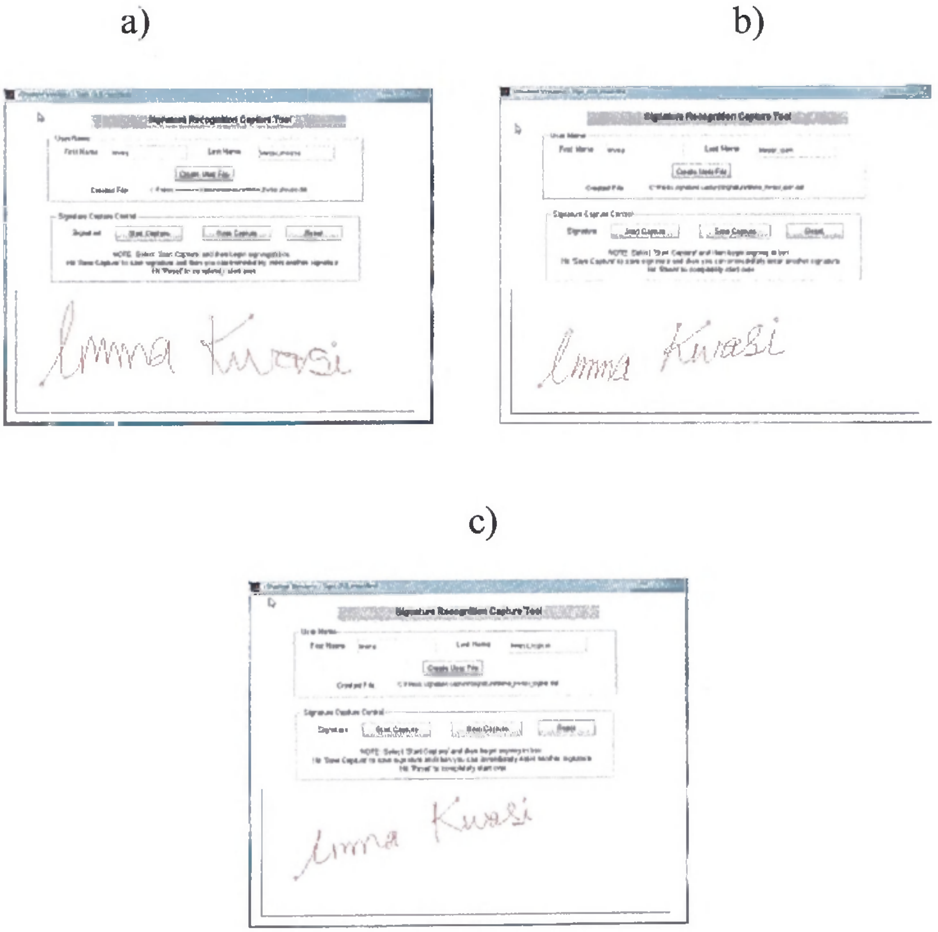


Figure 3.1.2.2 Snapshots of capturing handwritten signatures with the three selected input devices a) Signature captured with the mouse b) Signature captured with the I-Pen c) Signature captured with the IOGear.

3.1.3 LAYOUT OF THE SIGNATURE

The layout of the signature refers to the arrangement of the signature as the user signs their respective names in the interface provided for the data capture. The layout of the name signing includes the position of the signature, the rows and columns where the signature occupies, the flow of the signature and the appearance of the signed name.

Figure 3.1.3.1 shows the tool designed to display the handwritten signature capture. The "Browse" button allow user to select the kind of file s/he want to display. The display tool also has the button for displaying all signatures which allows multiple displays of the signatures as well as individual signatures. This feature allows for the careful study of individual signature as well as multiple signatures. One major aspect of the research is to study how consistent users are when signing their names. With this tool it is much easier identifying the difference among the respective signatures. Figure 3.1.3.2a, Figure 3.1.3.2b and Figure 3.1.3.2c shows an example of user's signature displayed with the mouse, the I-Pen and the IOGear respectively. Similarly, Figure 3.1.3.3a, Figure 3.1.3.3b and Figure 3.1.3.3c shows examples of the multiple display of handwritten signatures with the mouse, I-Pen and IOGear devices respectively.

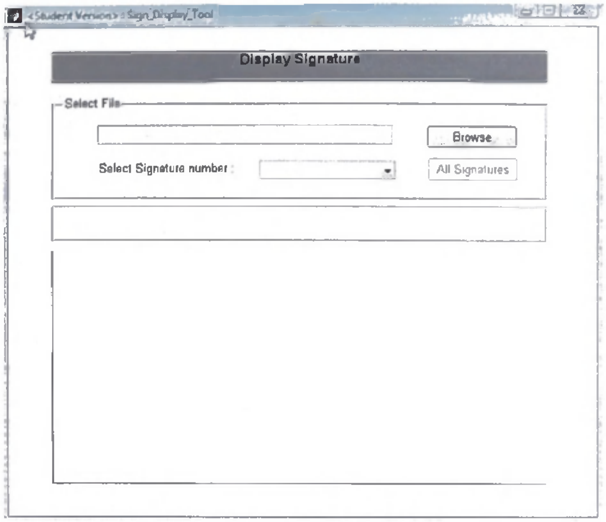

<Student Version> : Sign_Display_Tool
Display Signature
Select File
Browse
Select Signature number :
All Signatures

Figure 3.1.3.1 Snapshot of the displaying tool

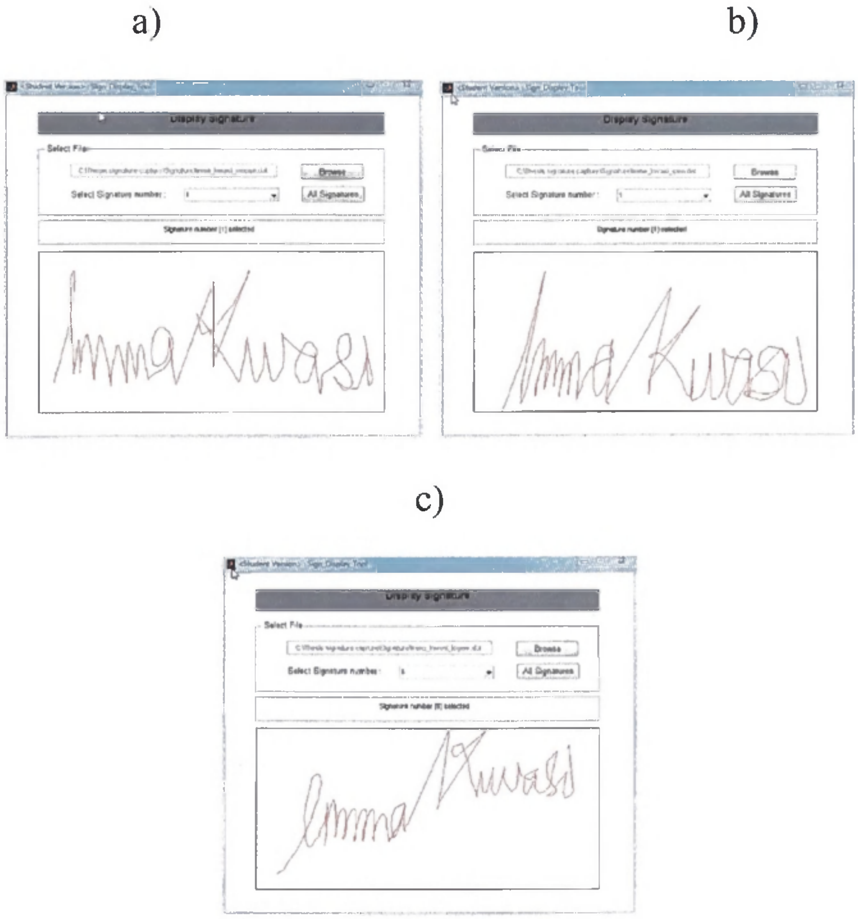


Figure 3.1.3.2 Snapshots of the display of the captured handwritten signatures with the selected input devices a) Display of signature captured with the mouse b) Display of signature captured with the I-Pen c) Display of signature captured with the IOGear.

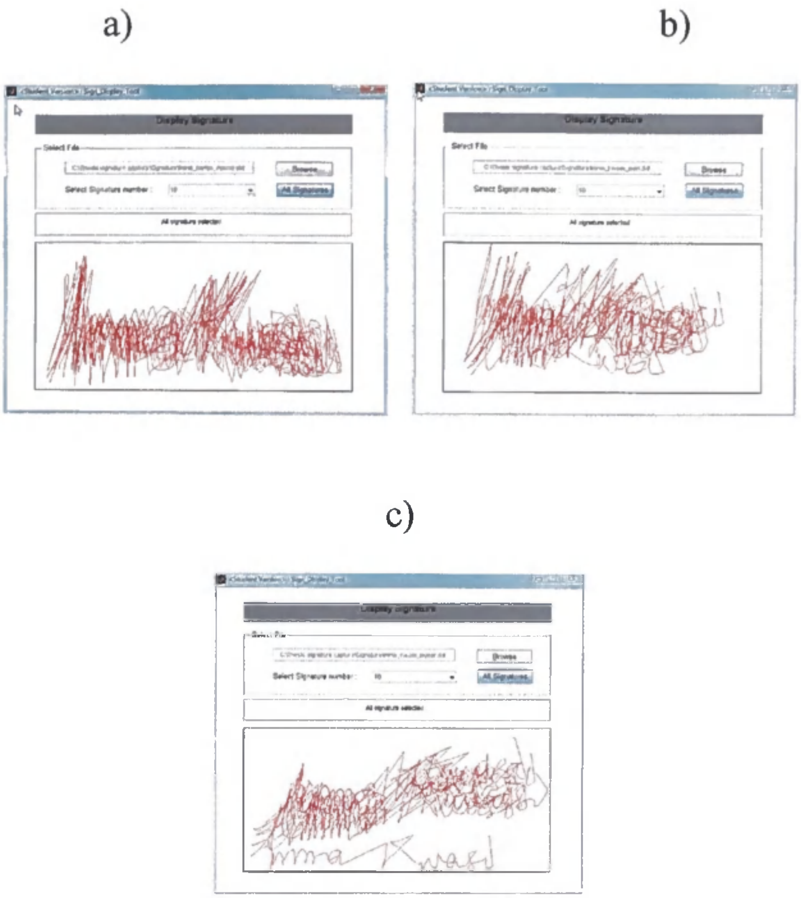
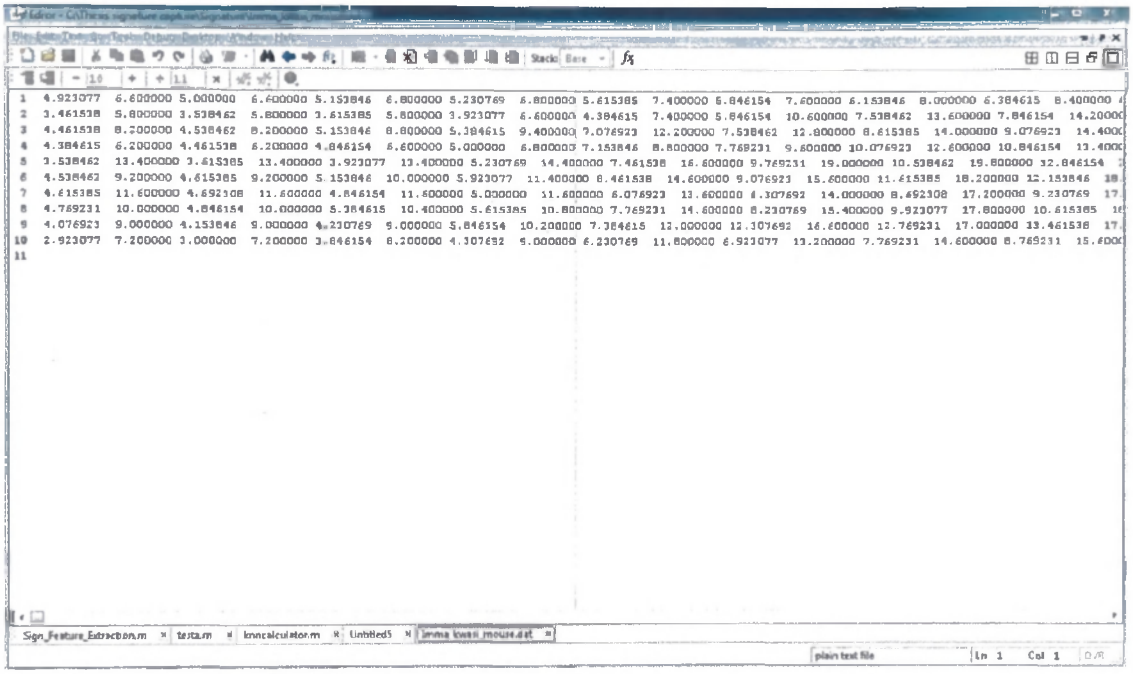


Figure 3.1.3.3 Snapshots of the display of multiple signatures from the selected input devices a) Display of ten (10) captured signatures from the mouse b) Display of ten (10) captured signatures from the I-Pen c) Display of ten (10) captured signatures from the IOGear.

3.1.4 SIGNATURE STORAGE AND RETRIEVAL

The captured signature for the selected input devices was stored in separate files. The signatures are stored in one file after each individual has signed their respective names number of times with the input devices. For example, user 1 signed his name with the mouse ten times. These ten signatures are stored in one file. The same procedure for the IOGear and the I-Pen input devices. The captured signatures that were described in sub-section 3.1.2 were stored in an *'x'* and *'y'* coordinates as shown in Figure 3.1.4.1 below. The first signature signed was stored in *'x'* and *'y'* format on the first row, the second signature was stored in *'x'* and *'y'* format on the second row and it continues to the tenth signature. The signatures that were stored in a ".DAT" format could be retrieved using MATLAB and any other software that accept this file extension. In the case of this research MATLAB was used to retrieve the signature for analysis. The signatures were retrieved in the *'x'* and *'y'* format, the same way it was stored.



```
4.923077  6.600000 5.000000  6.600000 5.153846  6.800000 5.230769  6.800000 5.615385  7.400000 5.846154  7.600000 6.153846  8.000000 6.384615  8.400000
3.461538  5.800000 3.538462  5.800000 3.615385  5.800000 3.923077  6.600000 4.384615  7.400000 5.846154  10.600000 7.538462  13.600000 7.846154  14.2000
4.461538  8.200000 4.538462  8.200000 5.153846  8.800000 5.384615  9.400000 7.076923  12.200000 7.538462  12.800000 8.615385  14.000000 9.076923  14.400
4.384615  6.200000 4.461538  6.200000 4.846154  6.600000 5.000000  6.800000 7.153846  8.800000 7.769231  9.600000 10.076923  12.600000 10.846154  13.400
3.538462  13.400000 3.615385  13.400000 3.923077  13.400000 5.230769  14.400000 7.461538  16.600000 9.769231  19.000000 10.538462  19.800000 12.846154
4.538462  9.200000 4.615385  9.200000 5.153846  10.000000 5.923077  11.400000 8.461538  14.600000 9.076923  15.600000 11.615385  18.200000 12.153846  18.
4.615385  11.600000 4.692308  11.600000 4.846154  11.600000 5.000000  11.600000 6.076923  13.600000 6.307692  14.000000 8.692308  17.200000 9.230769  17.
4.769231  10.000000 4.846154  10.000000 5.384615  10.400000 5.615385  10.800000 7.769231  14.600000 8.230769  15.400000 9.923077  17.800000 10.615385  1
4.076923  9.000000 4.153846  9.000000 4.230769  9.000000 5.846154  10.200000 7.384615  12.000000 12.307692  16.600000 12.769231  17.000000 13.461538  17.
2.923077  7.200000 3.000000  7.200000 3.846154  8.200000 4.307692  9.000000 6.230769  11.800000 6.923077  13.200000 7.769231  14.600000 8.769231  15.600
```

Figure 3.1.4.1 Snapshot of the ten signatures of the mouse input device of one user stored.

3.2 FEATURE EXTRACTION

Signatures captured and stored in the 'x' and 'y' format required further processing before classification will be possible. For further processing of these signatures, Discrete Fourier Transformation and normalization according to Rafiei's paper were implemented. To specifically define what Discrete Fourier Transformation and its implementation in Rafiei's paper, let's consider 'N' as points of an image of a discrete function X (n) = (x1(n), x2(n)) [89]. Using this function we can now define a discrete complex function u (n) as u (n) =x1(n) +jx2 (n). u (n) can be transformed into the frequency domain by the Discrete Fourier Transformation (DFT). The result could then be transformed back into the spatial domain through the Inverse Discrete Fourier Transformation (IDFT) [89, 82]. DFT and IDFT are defined respectively as;

$$a(k) = 1/N\sum_{N=0}^{N-1} u(n)\ ^{-j2\Pi kn/N} \quad k=-N/2,\ldots,N/2 \qquad u(n) = \sum_{N=0}^{N-1} a(k)\ ^{j2\Pi kn/N} \quad n=-N/2,\ldots,N/2$$

According to Jain [90], the coefficients of a(k) are called the Fourier Descriptors. They represent the discrete contour of the shape on a Fourier domain. Certain geometric transformations of the contour function u(n) can be related to simple operations in the Fourier domain. Transformation by 'u_0'affects only the first Fourier descriptor a(0), while the other Fourier descriptors retain their values. Scaling of the contour with a factor 'α' leads to scaling of the Fourier descriptors by 'α'. Rotating the contour at an angle of 'θ_0' yields a constant phase shift of 'θ_0' in the Fourier descriptors. Changing the starting point of a contour at 'n_0' position, results in a linear phase shift of $2\Pi n_0k/N$ in the Fourier descriptors [90].

On the other hand Rafiei [82], takes into account two boundary functions b_t=xt+jyt and $b_t'=x_t'+jy'_t$ (t=0,...., N-1). Computing the Fourier descriptor for both boundaries should solve the ambiguity of the Euclidean distance computed between the two boundaries. Rafiei's proposal was to obtain the Fourier descriptors for every shape boundaries. After obtaining the Fourier descriptors, you compute the fingerprint for every shape. The fingerprint is followed by similarity queries and for queries that use transformation in their expressions of similarities, should apply transformation to the index as necessary.

Rafiei's fingerprint computation involves transformations of the descriptors. First, B_0 is set to '0'. B_0 is the only descriptor that carries information about the location of the shape [82]. Next, the scale normalization is achieved by dividing every coefficient B_f by the amplitude of B_1, often called the fundamental frequency. After the normalization, B_0 is 0, so we do not need to store it, instead, the original value of B_0 before the normalization. The real and the imaginary parts of the initial value of B_0 represent the shift factors, respectively, along the 'X' and the 'Y' coordinates; the amplitude of the initial value of B_1 represents the scale factor. To totally get rid of B_1, which already has amplitude of 1 for all shapes, we do an additional normalization. We shift the starting point such that the phase of $B1$ becomes zero [82]. Rafiei's definition of his proposal is as described below. Given the Fourier descriptors ($B{-}M$, . . . , BM) of a shape, denote the real part of B_0 by sh_x, the imaginary part of B_0 by sh_y, the amplitude of B_1 by 'sc', and the phase of B_1 by 'p'. The shape description is defined as the sequence (sh_x, sh_y, sc, S_{-1}, S_2, S_{-2}, S_3, S_{-3}, . . . S_M, S_{-M}). Where $S_i = ((B_i - (sh_x + sh_y j))/sc) * e^{-ipj}$ (a complex number) for $i = -1, \pm 2, \pm 3, \ldots$ The Euclidean distance between two shape descriptions, irrespective of variations in location and size, can be computed as;

$$D^2(\mathbf{S}, \mathbf{S'}) = \sum_{f=-M, f\neq 0,1}^{M} |S_f - S_f|^2$$

Similar shapes often times have different size and orientation. According to Rafiei's paper [82], the Euclidean distance computed for two shapes was different when one of the shapes was rotated at a certain angle. A simple approach in removing this difference due to shifting, scaling and rotation was to normalize the Fourier descriptors before storing but there is no guarantee that the distance between the two shapes will be minimized and secondly there is no guarantee that normalization will always be the best option since shapes like '6' and '9' should not be treated as similar shapes.

To achieve a better classification, the signatures were analyzed as a raw data, Rafiei's phase shift and Rafiei's normalization formula. The second and the third normalization was implementation of Rafiei's [82] phase shift and normalization formula. The phase shift normalization is achieved by computing the phase of each signature which result in each signature having its first index to be zero and the second index been one. After which each of the signatures are multiplied by the phase shift factor.

The signature which was captured was read from file and converted to a one-dimensional complex numbers using the function *"xi + jyi"* and the Fourier transform algorithm. The Fourier transform algorithm is given by $X_k = \Sigma^{N}_{n=1}$ x(n)*exp(-j*2*pi*(k-1)*(n-1)/N), where 1 $\leq$k $\leq$N. The x-axes of the signature were represented as the real part and the y-axes represented as the imaginary part from the first point of the signature to the last point on the signature. The complex numbers were normalized using the basic method. The basic method was to take the first index of each respective Fourier transform signature and divide through the whole Fourier transform. The second normalization was the phase shift normalization where the first index of each respective Fourier transform signature was set to zero and the second index multiplied through the Fourier transform. Finally the signature was defined by the sequence described by Rafiei et. al [82].

3.3 FUZZY CLASSIFICATION

To aid in the classification of the signatures from the three selected input devices, the signatures under study went through six stages as displayed in Figure 3.1.1. The first stage was the preprocessing stage. Following the preprocessing stage was the transformation and normalization stage. During these stages inter-class and intra-class distances were computed. Inter-class distances were achieved by computing the distances between the testing signature and the training signatures from one person. Intra-class distances were also achieved by computing the distances between the testing signature and the training signatures from one person to another. These distances are displayed in Tables 4.2.1, Table 4.2.2, Table 4.2.3 and Table 4.2.44. These distances were used in computing the membership functions for each individual as discussed in section 3.3.1 below. After the membership functions have been derived for each person, and the cutoff threshold about 85% confidence level has been established, an individual's signature goes through two phase of classification using the fuzzy k nearest neighbor algorithm and fuzzy If-Then classifier.

The first phase of the classification narrows down the number of signatures that stands the chance of been classified as the accepted signature. That is when the signature of an individual is run against the signature of other people's signature, the distance between the selected testing signatures from an individual and the training signatures from other people's signature are computed. The k-NN classifier is implemented to select the best signature. The value of 'k' used in this research is '1'. At 1-NN classifier the best signature is selected. This process continues till all the signatures of an individual have had the opportunity to become the testing signature. These best distances from the classification are made available for the second and the final phase of the classification.

The second phase of the classification classifies the signature as either accepted signature or rejected signature based on the cutoff threshold established during the membership function generation. The classifier used at this phase of classification is fuzzy If-Then classifier. This classifier is a rule-base classifier as discussed in section 2.2.3.3.1. This classification phase implemented two rules, IF x < Cutoff Threshold THEN ACCEPT and IF x > Cutoff threshold THEN REJECT. That is if distance on the list from phase one falls below the cutoff threshold of that testing signature then ACCEPT signature otherwise REJECT signature.

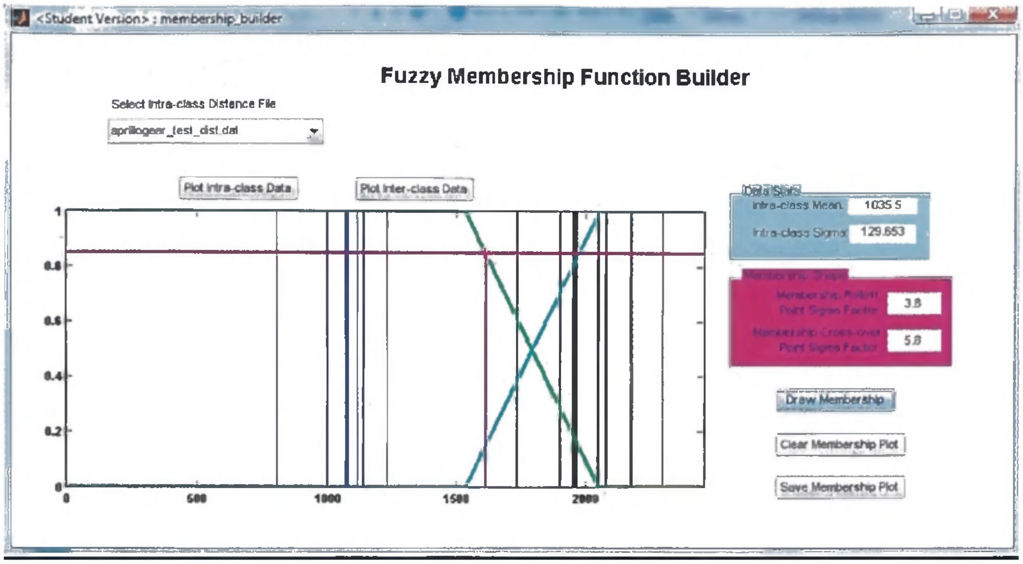
3.3.1 MEMBERSHIP FUNCTIONS

In reference to D. Driankov, H. Hellendoorn and M. Reinfrank, three things should be made clear about membership functions [79]. Firstly, precise membership degrees do not exist by themselves, but are only tendency indices that are subjectively assigned by an individual. Thus, the membership degree is not a primitive object; rather it reflects an ordering of the object of the universal set induced by the subsets. Secondly, the membership degrees are not absolutely defined but in most cases context dependent. Lastly, fuzziness differ from imprecision in that imprecision refers to lack of knowledge about a value of a parameter, example height, and is thus expressed as a crisp tolerance interval. This interval is the set of all possible values of a parameter. Fuzziness occurs when the interval has no sharp boundaries.

After normalization of the data, inter-class and intra-class variability of the signatures were computed. These variability computations were used in generating the membership functions as tabulated in Appendix A. The membership functions were used in determining the threshold at which a signature is to be classified by the fuzzy classifier.

The membership function is used in determining the threshold for an individual. This established threshold will be used by the fraud detection tool and the fuzzy "IF-THEN" classifier for determining the level of forgery and the probability of correct and incorrect classification.

The membership functions of each individual signature were derived. In determining the membership function of these signatures the intra-class variability distance computed were used alongside the inter-class variability distance. The mean of the Intra-class variability distances were computed to facilitate selection of the best cutoff for each signature. The mean, the upper limit of the standard deviation and the lower limit of the standard deviation of intra-class distances were computed. Intra-class distance were plot and the boundary of the acceptance was reduced or increased based on the distribution of the intra-class distances, the mean of the inter-class distance, the mean of the intra-class distances and the standard deviation. The "roll off point sigma factor" and the "crossover sigma factor" were varied in determining the final membership function as displayed in Figure 3.3.1.1. The membership functions for all the signatures for each selected input devices are displayed in Appendix A.



Fuzzy Membership Function Builder

Select Intra-class Distance File: aprilogear_test_dist.dat

Data Stats

Intra-class Mean: 1035.5

Intra-class Sigma: 129.653

Membership Shape

Membership Rolloff Point Sigma Factor: 3.8

Membership Cross-over Point Sigma Factor: 5.8

Figure 3.3.1.1 the membership function builder displaying the membership function of an individual.

3.3.2 COMBINED FUZZY NEAREST NEIGHBOR AND FUZZY IF-THEN ALGORITHM

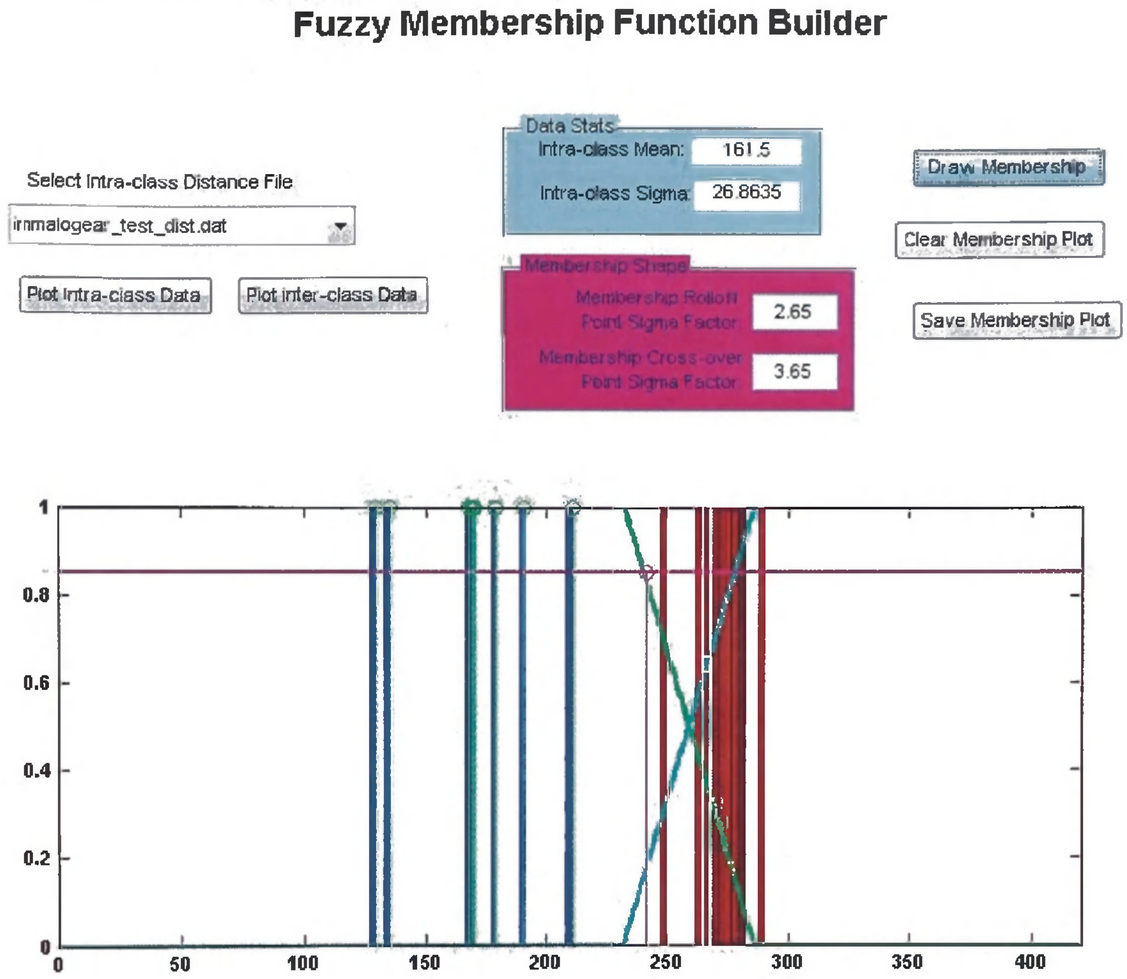
The last stage of this research work is the classification of signatures. To assist in classifying these signatures a combined fuzzy k-nearest neighbor algorithm and fuzzy If-Then algorithm were implemented. These two algorithms divide the classification phase into two as discussed in section 3.3. The first phase is the implementation of the fuzzy k-NN algorithm and the second phase is the implementation of fuzzy If-Then algorithm. The value for 'k' selected for this research was '1'. The main reason for selecting k=1 is that, this research want to make sure that the distances of unauthorized signatures (signatures from other people) does not fall below the cutoff threshold for an individual. Since the smaller the distance the better the chances of a signature been granted as accepted, this research wants to see to it that the smallest distances are selected for the final classification when the signatures from an individual is tested against other people's signatures. This will result in selecting a signatures of other people that stands a higher chance of been accepted. This also tests how efficient the system is in term of classifying signature that belongs to others and not the original user.

Preceding the classification of the signatures using the combined fuzzy k-NN and fuzzy If-Then classifier, the threshold for each individual signature was derived from the membership functions. The threshold for an individual was computed by using the membership function generated using the membership function builder tool as displayed in Figure 3.3.1.1. The "roll off point sigma factor" and the "crossover sigma factor" were varied in determining the final membership function. After the best membership function is determined, 0.85 (85%) degree of membership is selected on the y-axis on the membership builder as shown in Figure 3.3.1.1. A horizontal line is drawn parallel to the x-axis at the 0.85 degree of membership. The point at which the horizontal line meets the inter-class distances part of the membership function (which in our case is the green colored line), the closest minimum values and the closest maximum values are interpolated to determine the correct point of contact. The interpolated value is traced down to the x-axis. The point at which the interpolated value meets the x-axis becomes the threshold of that membership function. The thresholds of all the membership function at 0.85 degree of membership is displayed in subsection 4.2.

The derived cutoff threshold is used during the two classification phase of this research. Fuzzy k-NN algorithm is implemented during the first phase of the classification process. The value of 'k' as discussed before is '1'. With individual signature to other people's signature classification, one signature is selected from an individual's signature to represent the testing signature. The testing signature is run against other people's signature with k=1. 1-NN algorithm is used to classify the signature and the best signature is selected. The second testing signature is run against the other people's signature; 1-NN is used to classify the best signature. This procedure continues till all the testing signatures take turns. The best classified signatures using 1-NN algorithm are used in last classification process.

The final stage of classification is designed to determine if the nearest neighbor result is 'close enough' to the actual training samples. Since the idea of 'close enough' can be addressed by fuzzy logic, we use a fuzzy "If-Then" classifier for this final stage. According to Ludmila [84], there are three popular acronyms for fuzzy "If-Then" systems namely, SISO- Single input single output (n=c=1), MISO-Multiple input single output (n>1, c=1) and MIMO- Multiple input multiple output (n>1, c>1). The fuzzy "If-Then" system used for the purpose of this research was SISO. This classifier used two (2) fuzzy "If-Then" rules. The rules are; a) IF x < Cutoff (85%) THEN "Accept". b) IF x > Cutoff (85%) THEN "Reject". The Cutoff (85%) denotes the 85% degree of membership for every other individual and for a specific input device as established in 'i' above. The 'x' denotes the distances computed for each signature. The "Accept" and "Reject" denote two major fuzzy classes. The "Accept" class is the class where that signature is classified as the correct signature and the "Reject" class is the class where that signature is classified as incorrect signature. The rules have an antecedent part and the consequent part. The antecedent part is the "IF" part of the rule. The consequent part is the "THEN" part of the rule.

To confirm whether the classified signatures were classified correctly by the combined fuzzy k-NN and fuzzy If-Then classifiers, the distances of the classified signatures were mapped into the membership function of that individual at the same cutoff threshold (85% or 0.85 confidence level). During this mapping procedure, if signature is classified as "ACCEPT" there is a higher expectation that the mapping should fall below the 85% confidence level. Any other result will render the classification as faulty. On the other hand, if a signature is classified as "REJECT" there is a higher expectation that the mapping will be above the 85% confidence level. The membership function builder was used during the mapping process. As displayed in Figure 3.3.2.1, the magenta vertical line represent the 85% confidence level while the green vertical line with circle at the top represent the classified distances. In the diagram, all the best selected signatures that were classified fell below the 85% confidence level. These signatures were also classified as "ACCEPT". This confirms the fact that the combined fuzzy k-NN and the fuzzy If-Then classifier classified the signatures correctly.


Fuzzy Membership Function Builder

Select Intra-class Distance File: irnmalogear_test_dist.dat

Data Stats
Intra-class Mean: 161.5
Intra-class Sigma: 26.8635

Membership Shape
Membership Rolloff Point Sigma Factor: 2.65
Membership Cross-over Point Sigma Factor: 3.65

Figure 3.3.2.1 mapping of classified distances into the membership confidence level.

4. EXPERIMENTAL EVALUATION

This section is devoted in describing how the experiment was performed. The first section deals with how the ideas already discussed in the previous sections and their respective sub-sections were implemented in this research work. The methodology sub-section describes the various methods that were used in accomplishing this research work. Sub-section 4.3 will be focused in discussing the outcome of the research work.

This will detail the findings during the experiment and the following sub-section will be the recommendation.

4.1 IMPLEMENTATION OF EXPERIMENT

In addition to correct versus missed classification, fraud is a critical component of an online user authentication system. To support this testing, a fraud detection tool was also designed in testing one's signature against a forged signature. This test was to experiment how easy it will be for others to forge someone's signature. The membership functions for each signature were determined based on the intra-class variability, inter-class variability, the mean and standard deviation of the intra-class distances. This research aimed at using the fuzzy nearest neighbor classifier and the fuzzy If-Then classifier to classify signatures as accepted or rejected based on the cutoff threshold set by each membership function. A confusion matrix was generated based on the total number of signatures that will be classified as accepted or rejected.

4.2 EXPERIMENTAL METHODOLOGY

The algorithm used in the distance computation was the Euclidean distance. The minimum distance from the testing data to the training data was selected as the closest signature to the testing signature. The minimum distance, testing signature and the closest signature are displayed on a graphical user interface (GUI).

Intra-class variability was achieved by selecting one signature out of the ten signatures signed by an individual, computing the distances among them and selecting the nearest neighbor (that is the smallest distance among the computed distances). The selected signature is considered as a testing signature. Euclidean distance is used in determining the difference between the selected signature and each other signature which belong to the same individual. These distances are considered as an intra-class distances. Inter-class variability was also achieved by computing the distance between individual signatures and all the other signatures. Inter-class distances were achieved the same way as intra-class distance but this time the test signatures is selected from one person's signatures and that signature is computed against the other signatures signed by other people. Both inter-class and intra-class distances were used in generating the membership function for individual signatures. Table 4.2.1 displays the averages of individual intra-class distances for the three input devices. Table 4.2.2 Table 4.2.3 and Table 4.2.4 are the representation of both the minimum and maximum values of the inter-class distances for an individual and their respective class ID-the class of signature which that particular signature is closest

Intra-class distances						
Signature number	Mouse		I-pen		IOGear	
	Avg dist	Std dev	Avg dist	Std dev	Avg dist	Std dev
1	976.84	281.1537	754.95	232.2433	600.5235	244.6839
2	242.7985	55.99948	422.3879	198.4088	258.3104	92.02056
3	357.9486	64.69382	597.9911	119.9594	442.6625	56.85671
4	314.8965	53.4182	453.7011	99.52159	297.2333	258.9801

5	161.3141	19.02174	341.3636	168.077	166.1796	27.7162
6	132.9505	18.66152	166.8636	11.68568	96.87547	11.606
7	230.4458	29.40907	211.6032	66.85878	311.5045	60.61674
8	68.97792	25.82835	28.68915	9.216152	15.49839	3.522647
9	162.0784	14.23991	214.829	20.81036	89.79882	7.220302
10	330.6598	35.80362	356.8488	93.31348	152.8669	14.61123
11	347.0038	193.8009	318.3228	189.8632	171.3164	34.99231
12	205.4191	36.80374	459.14	399.8182	213.5793	69.38572
13	261.9676	22.37354	232.7367	50.87639	163.4476	17.7289
14	714.3233	40.42963	601.8169	74.89415	481.7584	54.0957
15	634.6093	180.0066	1006.86	1184.58	194.272	28.5441
16	812.25	198.2066	763.33	148.1616	402.7332	185.3654
17	323.6657	58.72613	278.2379	36.64991	115.4254	32.00235
18	47.61167	7.607381	99.12318	26.26582	45.46018	9.808856
19	1813.48	774.0532	2890.73	412.6978	882.6667	158.5063
20	410.5535	53.35477	312.0249	30.97434	161.5114	28.22337
21	586.4952	118.6503	622.45	176.3431	300.3291	55.80808
22	545.8797	108.8269	474.7517	107.9708	400.7677	55.05775

23	396.5599	21.70339	390.0361	33.97668	294.4926	19.97413
24	587.7128	138.4473	354.7865	44.60859	128.3851	29.92142
25	299.6555	33.8222	335.6526	82.78047	138.9207	110.644
26	410.5824	75.41833	346.4928	92.17792	128.4853	12.14092
27	273.2237	29.46187	845.98	113.3972	523.1273	286.3812
28	330.4019	47.25585	196.9553	35.97059	103.3808	10.1928
29	524.9914	63.20128	1694.03	832.6339	299.9907	38.79442
30	150.8761	11.72845	166.1734	30.61687	615.98	615.0255
31	152.7316	27.01597	131.8526	33.99567	227.9049	54.47186
32	516.7811	40.22107	357.5568	66.64868	247.8895	47.88244
33	856.94	150.4608	1490.03	285.1031	488.9665	71.26283
34	533.0891	151.0959	1033.74	140.951	288.4144	29.30075
35	231.8747	28.19438	313.7015	53.91496	214.6404	25.58472
36	39.07557	11.5566	93.91781	27.19615	39.41655	53.362
37	488.9226	63.33006	901.36	1010.849	231.3071	12.2355
38	424.8651	147.8592	345.4895	71.08987	226.5471	51.30099
39	305.8163	63.67762	246.4977	66.26401	835.1714	761.2583
40	13.33018	3.351396	12.51569	2.157992	19.15345	44.61057
41	187.8019	76.3521	217.9066	49.97152	130.8533	44.10461
42	242.7057	23.88878	252.4915	40.31576	109.8378	6.185982
43	184.2938	12.70661	197.783	63.20222	111.6077	71.6672
44	270.3989	31.30705	197.8433	35.64408	236.46	275.2272
45	418.2961	32.51304	448.7443	49.65302	399.1296	78.35486

46	1179.13	99.5739	502.0904	81.30515	307.5964	16.74139
47	255.2546	14.50229	191.9612	28.45202	185.9585	63.46953
48	39.71099	6.843195	29.39927	4.858406	18.71645	6.500808
49	954.53	186.2878	2460.06	711.3035	536.4612	111.1938
50	277.4964	48.10367	420.1206	196.5694	134.5766	48.00354
51	335.7862	56.53196	320.9802	108.913	366.2636	322.6068
52	54.43291	18.56674	53.69814	42.93042	12.89848	8.899692
53	40.81659	13.35469	80.53774	39.11806	79.29364	29.35402
54	97.93932	29.61014	74.75158	58.86106	35.87567	11.87396
55	17.03427	14.77895	10.97432	2.423008	12.39051	3.805665
56	497.6208	143.8907	500.3929	60.62207	287.9638	44.75055
57	320.2676	28.25281	355.4379	65.50461	342.1584	22.46694

Table 4.2.1 Intra-class distances of the three input devices.

Inter-class distances						
Signature	Min		Max		Average for Min	
	value	Class ID	value	Class ID	Avg dist	Std dev
1	245.9103	17	704.7728	33	500.9597	169.0061
2	271.7139	43	332.4395	6	290.6871	23.16107
3	357.7254	51	557.2359	22	452.1453	77.84498
4	289.5394	35	633.3915	45	327.7477	97.15951

5	237.5835	54	304.1445	24	258.1645	22.949
6	286.3546	11	332.836	52	306.6578	16.21694
7	316.4095	42	535.4497	29	407.8519	93.30558
8	321.5604	21	355.9669	13	339.0576	7.802518
9	329.0679	36	361.5454	12	344.9564	13.51064
10	238.8097	5	272.9037	20	254.5449	12.72155
11	272.3133	24	518.5799	7	356.2081	93.10259
12	303.3137	46	366.7193	55	323.5729	21.91873
13	295.9759	4	356.9502	9	325.5764	23.3637
14	477.6176	39	653.3472	44	534.1108	54.91601
15	254.0242	26	301.6534	5	281.8438	18.43509
16	421.721	41	865.2794	51	624.3938	156.4271
17	244.3669	53	344.6157	46	261.2043	30.12028
18	362.4667	44	409.5839	47	382.2335	11.79636
19	1028.4	49	1842.8	30	1359.938	278.7216

20	264.1742	15	290.4918	15	273.4224	11.48374
21	321.0573	28	513.5916	11	372.542	63.23478
22	450.8765	16	596.7104	4	506.0541	82.67356
23	307.4183	12	456.1565	38	356.435	48.13489
24	272.2505	2	318.0168	35	296.2204	19.82515
25	247.3264	1	340.8608	28	287.6833	30.95736
26	248.4684	25	352.1542	42	275.5655	34.56778
27	398.1395	30	966.1105	49	550.1164	205.3984
28	319.783	7	342.6864	17	328.1544	12.91461
29	385.7433	50	546.2382	3	439.9407	54.46668
30	389.1	29	3351.7	39	1614.63	825.7637
31	362.4764	18	639.7134	14	498.6793	123.1226
32	371.5934	45	496.6568	21	403.6718	36.90973
33	412.3896	27	712.5904	16	558.8654	108.3348
34	326.2782	8	427.786	38	369.23	48.68423
35	288.4788	56	326.5549	2	305.8277	12.82486

36	328.8611	34	463.7773	41	434.453	33.14972
37	333.8977	57	371.2606	40	349.9216	17.48384
38	287.143	6	434.6876	43	337.1847	48.95874
39	463.6	22	3870.5	39	1366.386	1348.156
40	330.9263	55	376.2961	53	360.2365	11.62966
41	421.639	33	478.0371	50	443.4782	30.03835
42	316.0798	47	355.0096	8	335.9132	13.51349
43	270.0648	20	439.423	56	390.3119	69.05639
44	358.2569	3	694.7651	1	407.6687	102.7698
45	368.3453	31	636.0457	31	501.5716	105.8313
46	301.1165	52	346.3566	26	322.2947	17.06435
47	308.1428	23	424.5734	34	331.926	37.97526
48	331.7621	40	369.9878	37	351.4371	9.958358
49	749.0113	14	972.1402	19	831.4056	105.5373
50	377.9641	32	478.8353	57	409.3119	57.89114
51	342.2716	37	958.787	27	430.8511	181.1532

52	298.9694	13	336.7442	25	317.2012	8.899226
53	243.1843	10	384.4838	54	327.9588	19.1149
54	230.6579	54	391.0025	18	301.9958	28.04162
55	329.2751	9	368.5028	48	350.6551	8.556837
56	287.8854	38	449.5136	23	370.742	63.55512
57	333.2384	48	487.0369	32	415.7816	68.72621

Table 4.2.2 Inter-class distances for IOGear input device.

Inter-class distances						
Name	Min		Max		Average for Min	
	value	Class ID	value	Class ID	Avg dist	Std dev
1	549.6	21	1472.6	51	898.04	318.1043
2	472.7	25	1094.3	21	641.08	197.0269
3	514.2677	6	927.3358	12	703.3914	145.564
4	525.7156	45	784.5208	50	616.8545	81.30528
5	507.8902	54	751.0231	41	567.5064	76.21384
6	513.4467	24	575.4621	7	524.7275	19.25468
7	459.4369	53	579.7489	24	504.7619	39.69134

8	487.5344	57	533.3299	28	507.5591	19.10493
9	597.1518	55	649.166	53	618.7353	18.74595
10	564.2343	47	633.0878	18	583.3752	30.65936
11	497.4086	48	773.2889	38	551.1021	85.36058
12	951.6287	37	988.916	22	942.6234	85.73486
13	477.8228	2	559.2653	17	522.7976	35.45299
14	588.2288	30	884.982	3	738.2988	135.8266
15	635.7	36	3522.9	29	1018.83	884.478
16	660.4	43	1625.9	33	1155.06	365.1114
17	481.6008	20	560.5958	6	502.4614	27.94056
18	547.3058	38	636.1788	54	610.9558	22.06636
19	3191.3	29	8405.8	19	5416.4	2173.816
20	481.2251	13	518.6392	48	497.078	13.00102
21	547.5	18	1127.3	1	720.3	191.2528
22	714.5706	31	999.6627	2	808.4662	134.6343

23	520.062	32	643.0447	9	570.0688	47.47708
24	511.6088	5	584.7726	57	538.2348	30.57166
25	466.3422	7	617.521	40	505.3962	48.16931
26	442.9623	52	675.5873	30	515.3224	64.96595
27	4137.6	49	4278	49	4182.94	53.77108
28	491.098	8	547.39	13	519.3455	18.97959
29	2591.2	33	4010	27	2652.44	709.6643
30	579.5188	35	691.8815	43	613.4939	47.51047
31	666.6706	16	755.2785	56	696.8753	40.20616
32	517.6615	39	598.8082	42	534.055	24.94734
33	2132.1	34	3024.8	15	2695.28	498.9396
34	1180.1	12	1569.9	16	1311.7	145.308
35	573.7175	10	699.5731	36	617.602	37.92459
36	615.6488	56	705.0168	52	651.7035	18.04534
37	880.1	51	7395.5	19	3354.26	2653.646
38	545.1867	42	778.3205	4	635.8749	89.39675

39	517.0492	3	667.4597	26	566.0688	51.23596
40	598.1588	9	623.0084	10	609.6042	6.090033
41	637.7829	15	751.278	31	680.7823	48.40026
42	541.9492	4	603.1967	55	568.1164	29.76257
43	638.3776	41	694.7779	35	642.844	53.95123
44	448.2729	26	502.6511	20	470.6685	30.50226
45	520.3508	23	822.4134	14	649.5033	127.8571
46	554.2145	50	769.111	11	605.3793	70.36209
47	562.9292	46	667.2279	39	595.7757	37.2697
48	491.3041	28	532.5851	8	507.4162	8.556499
49	3599.5	19	5103.1	37	4345.06	1172.566
50	554.0645	1	803.1787	45	628.3962	78.78014
51	781.8	22	1509	34	947.41	284.8987
52	426.7584	52	720.7782	5	452.5607	70.24668
53	457.7265	44	654.9275	47	551.8184	44.34866
54	500.1879	11	639.7029	23	554.3176	28.21375

55	594.9927	14	616.4235	25	602.9053	5.503736
56	604.1811	40	766.527	46	635.2947	60.69765
57	481.6603	17	598.7643	32	508.3245	38.89861

Table 4.2.3 Inter-class distances for I-Pen input device.

Inter-class distances						
Signatures	Min		Max		Average for Min	
	value	Class ID	value	Class ID	Avg dist	Std dev
1	859.1	15	2152.5	46	1456.17	448.2819
2	400.8895	4	446.2594	44	415.2562	22.5168
3	396.334	38	845.5298	14	516.9222	169.8439
4	397.1767	3	502.4488	31	431.7648	37.86004
5	425.1295	45	487.5976	53	447.8711	22.59292
6	376.8274	28	437.8439	48	399.6708	22.521
7	332.6666	7	448.306	57	374.6682	45.81194
8	338.2489	7	404.0987	47	366.307	18.24688
9	478.5253	43	518.5313	28	503.6848	17.06355
10	389.2001	42	522.8746	50	441.2778	45.82539

11	427.3727	5	723.8972	29	469.1865	90.37994
12	415.195	27	470.3042	52	435.4352	20.15536
13	405.2296	30	439.0121	2	415.1652	15.84597
14	738.8273	33	886.9738	34	782.7167	45.66354
15	792.3	14	1497.4	49	1106.93	244.9806
16	640.9	29	1803.6	33	1341.14	455.756
17	444.3038	55	585.9947	41	480.1861	53.63304
18	448.4345	24	480.1304	5	464.1114	9.393112
19	2689.2	49	5395.9	46	3522.58	1239.745
20	458.2826	40	592.1686	45	515.5289	48.35559
21	565.6	32	1006.1	15	731.32	174.081
22	493.6088	25	960.3069	24	720.6028	162.8878
23	463.3993	50	640.6175	25	552.026	71.90642
24	445	39	1000.5	21	676.92	173.782
25	486.9046	9	648.4871	32	590.3173	67.29615
26	384.4574	48	692.3607	11	514.7849	99.88856

27	414.1847	35	447.8759	7	431.5176	14.02156
28	357.1953	57	521.7173	10	406.4827	61.60349
29	604.8414	21	740.4655	56	675.7303	48.15372
30	404.6816	53	467.7245	12	434.1712	21.01912
31	454.0883	18	514.2592	9	477.63	22.06582
32	532.7457	51	674.0345	26	600.3272	64.04339
33	711.6	16	1868.8	46	1296.33	477.6433
34	456.6987	31	908.1435	38	609.1537	148.7027
35	406.8654	13	466.2935	30	420.1366	17.60676
36	506.6994	22	575.8123	17	545.8075	16.85797
37	525.9715	41	639.9388	23	570.0046	45.27571
38	393.9762	44	922.7945	22	534.1343	160.9973
39	444.4418	17	560.3447	36	466.0244	34.80592
40	456.9604	34	477.3475	18	465.469	4.881381
41	518.2335	36	588.4689	20	541.1797	39.93684
42	385.1859	26	430.8545	6	405.9427	24.20968

43	466.539	23	501.9252	4	480.2186	16.17019
44	389.7435	10	447.8536	27	412.3511	23.82053
45	422.4341	12	592.9497	51	503.0183	65.59271
46	3906.3	19	4052.2	19	3960.63	56.43336
47	378.593	6	426.8837	42	396.9256	16.8002
48	378.9461	47	437.959	7	418.8933	11.99347
49	1014.2	1	1549.2	16	1250.92	165.0913
50	459.0706	20	544.786	39	480.3765	27.32684
51	532.3458	56	603.4409	37	561.6741	33.47313
52	344.7325	54	473.9435	40	392.2202	23.41221
53	402.2498	2	495.9542	43	446.8474	22.65635
54	340.3652	8	403.6137	8	359.9274	10.23768
55	432.4911	11	460.2157	35	444.6095	15.87139
56	529.5766	37	803.2975	3	643.1997	103.4689
57	349.2728	52	456.3231	55	384.0754	35.05893

Table 4.2.4 Inter-class distances for Mouse input device.

4.3 TEST PROCEDURE

To achieve the main objective of this research work, the following procedures were adapted in classifying the signatures under study:

1) Selecting training data set and testing data set: The training data set and the testing data set are selected based on the kind of testing that was performed. When the test is performed on an individual's signature and other person's signature, the training data set becomes the signatures of other people while the testing data set becomes the signature of that individual. When the test is performed on an individual's signature alone, the training set becomes the signature of that individual less one signature and the testing data becomes one of the individual's signatures. This continues until all the signatures of that individual have the opportunity to become the test signature (that is it runs through a loop until the end of the signatures).

2) Creating the confusion matrix: A confusion matrix is a matrix that contains all the information about a classification that is performed by a classification system. Our two class matrix in Table 4.4.1, Table 4.4.2 and Table 4.4.3 for IOGear, I-pen and the mouse input devices respectively are the results of the classification from the respective devices. Table 4.3.1 below shows an example of how the confusion matrix was generated for this research work. The letter 'a' is the total number of **incorrect** signatures that were **accepted**, 'b' is the total number of **incorrect** signatures that were **rejected**, 'c' is the total number of **correct** signatures that were **accepted** and 'd' is the total number of **correct** signatures that were **rejected**.

In determining the performance measure of this study, the confusion matrix was used in computing the Recall or True Correct (TC), False Correct rate (FC), True Incorrect (TI), False Incorrect (FI) and the Precision (P). TC is the proportion of the correct cases that were correctly classified as "Accept" (TC= c/(c+d)). FC is the proportion of the incorrect cases that were incorrectly classified as "Accept" (FC=a/(a+b)). TI is the proportion of incorrect cases that were correctly classified as "Rejected" (TI=b/(a+b)). FI is the proportion of the correct cases that were incorrectly classified as "Reject" (FI=d/(c+d)). P is the proportion of the predicted correct cases that were correctly classified as "Accept" (P=c/(c+a)).

Actual/Predicted	Accept	Reject
Sig.No -> Others	a	b
Sig.No -> Sig.No	c	d

Table 4.3.1 the structure of a confusion matrix.

***3) Correct and incorrect classification test*:** Correct and incorrect classifications are performed after the fuzzy classifier has classified the data into the two main classes and the confusion matrix has been generated. During the classification, the combined k-NN and fuzzy If-Then classifier classified the data into an "Accept" class or a "Reject" class. The total numbers of both classes were recorded into the confusion matrix tables in Appendix D. The confusion matrix tables were separated into two tables. The first table represents all the results of an individual to other people signatures where as the second table represents the results of individual's signatures against their own signatures. The total number of "Accept" and "Reject" in each table was computed. These values were used to form the overall confusion matrix for the respective input devices. The confusion matrix was used to determine probability of correct and incorrect classification. For each input device the TC, FC, TI, FI and P were computed. For better performance the geometric mean (that is the square root of the product of TC and P which can also be computed as the square root of the product of TC and TI) were computed for each of the input devices. True Correct (TC) represents the *correct classification* where as False Incorrect (FC) represent *incorrect classification.*

***4) Using the fraud detection tool*:** The fraud detection tool was designed to determine if an individual can forge someone's signature. This test was achieved by designing another tool as displayed in Figure 4.3.1 below. Firstly, the signature to be forged is loaded on the GUI. Secondly, the individual is asked to forge the displayed signature. The forged signature is saved into a folder labeled Forged Signature. Finally, the forged signature is loaded onto the fraud detection tool where the distance is computed and displayed on the GUI. If the distance computed is smaller than the cutoff distance for that particular individual then that signature is rejected otherwise that signature will be accepted.

In addition to the two major conditions is the individual fraud detection tool. This tool was designed to test, on the individual basis, how other people might be able to forge someone else signature. As displayed in Figure 4.3.1, the forger is presented the signature of the person he is to forge. After forging the signature, the distance between the forged signature and the real signature was computed. Moreover, the three normalizations were performed on individual signatures. The distance between the forged signature and the real signature was compared to the cutoff point established for that individual during the membership function building stage. If the distance happens to fall below the cutoff point, then the signature is considered accepted otherwise the signature is considered rejected.

For the purpose of this research the rate of fraud was tested by using the fraud test tool described above. The fraud test was performed on a selected number of signatures from the three input devices. The total number of attempts and the total number of correct and incorrect classification were recorded. The values were used in determining the rate of fraud for each input device. For the purpose of this test twenty signatures were selected at random from each input device. The twenty signatures from each input devices are considered as the *total number of attempts*. The total number of the signatures that are classified as correct are also considered as *total number of success*. The rate of fraud will be the ratio of the *total number of success* to the *total number of attempts*.

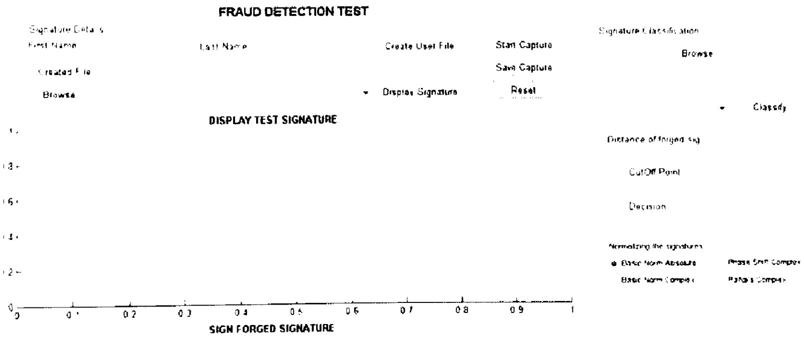


Figure 4.3.1 Fraud testing tool

4.4 RESULTS

On the basis of different trials and many variants of the experimental investigation such as the classifier used, generating the individual thresholds for each signature, normalization of the signature and the descriptors used, the following results were obtained which reflects the conclusion is section5. The two major conditions studied were as follows; firstly, the intra-class mean and the standard deviation for only the raw data were used in determining the cutoff point for each respective signature, and secondly the individual mean (that is raw data, Rafiei's phase shift and Rafiei's normalization formula) were used in computing the cutoff point for the signature.

The tables provided below show the results of the analysis of the three input devices in a confusion matrix format. The input devices were analyzed by using 85% confidence level. The signature column of the table display how the confusion matrix was achieved. From Tables 4.4.1 through to Table 4.4.3, "1->others" is a representation of an individual's signature as classified against other signatures. Like "1->others", "1->" is also a representation of an individual's signature as classified against his/her own signatures. The second column is the threshold for classifying the signatures at 0.85 degree of membership. The third column represents the number of signatures that were classified as correct. The fourth column displays the total number of signatures that were rejected. The last column is the total number of signatures classified for that particular individual. Table 4.4.1, Table 4.4.2 and Table 4.4.3 are the classification results for IOGear, I-pen and the mouse input devices respectively.

To demonstrate the classification of the signatures using the respective input devices, Figure 4.4.1, Figure 4.4.2 and Figure 4.4.3 display examples of the signatures as classified for IOGear, I-Pen and the mouse input devices respectively. The system must make a decision to either accept or reject the signature (when the testing signature is tested against the training data set). Examples of incorrect classification when test signature is tested against the person's own signatures are shown in 'a' and 'd' of Figure 4.4.1, Figure 4.4.2 and Figure 4.4.3. Examples of incorrect classification when the test signature is tested against another person's signatures are shown in 'c' and 'f' of Figure 4.4.1, Figure 4.4.2 and Figure 4.4.3. Finally, examples of correct classification when the test signature is tested against the person's own signatures are shown in 'b' and 'e' of Figure 4.4.1, Figure 4.4.2 and Figure 4.4.3. Incorrect classification when the test signature is tested against the person's own signatures is due to how variable signatures are from the same person signatures after being signed multiple times. A different approach could be implemented to capture the signature. How the signatures are captured greatly affects the analysis and classification of the signature. It was also observed that period ('.') normally found on top of letters such as 'i' and 'j' were displayed different from usual period. Additional research is needed to determine how these characters should be handled. This could also be attributed to the fact that, the average overall speed during the capturing of the signature vary greatly from one instance to another irrespective of whether the signature is been produced by a person's own or forging another person's signature. We cannot overrule noise as one of the factors that reduced the performance of this experiment. The correct classification was classified correctly as seen in the examples.

Signature	Cutoff(85%)	No. Accepted	No. Rejected	Total	(TC/total)*min
1 -> others	280	0	11	11	0
1 -> 1	280	3	8	11	2.45
2->others	315	0	12	12	0
2-> 2	315	10	2	12	7.5
3 -> others	390	0	12	12	0
3 -> 3	390	2	10	12	1.5
4 -> others	300	0	12	12	0
4 -> 4	300	10	2	12	7.5
5 -> others	230	0	10	10	0
5 -> 5	230	9	1	10	8.1
6 -> others	200	0	10	10	0
6 -> 6	200	10	0	10	9
7 -> others	289	0	9	9	0
7 -> 7	289	3	6	9	3
8 -> others	260	0	10	10	0
8 -> 8	260	10	0	10	9
9 -> others	170	0	10	10	0
9 -> 9	170	10	0	10	9
10 -> others	220	0	9	9	0
10 -> 10	220	9	0	9	9
11 -> others	285	0	11	11	0
11 -> 11	285	10	1	11	8.18
12 -> others	252	0	11	11	0
12 -> 12	252	11	0	11	9
13 -> others	460	0	14	14	0
13 -> 13	460	6	8	14	3.857
14 -> others	245	0	9	9	0
14 -> 14	245	9	0	9	9
15 -> others	410	0	12	12	0
15 -> 15	410	6	6	12	4.5
16 -> others	240	0	10	10	0
16 -> 16	240	10	0	10	9
17 -> others	1010	0	13	13	0
17 -> 17	1010	9	4	13	6.23
18 -> others	241	0	10	10	0
18 -> 18	241	10	0	10	9
19 -> others	293	0	10	10	0
19 -> 19	293	6	4	10	5.4
20 -> others	480	2	10	12	1.5
20->20	480	11	1	12	8.25
21->others	319	3	9	12	2.25
21->21	319	12	0	12	9
22 -> others	260	0	10	10	0

22 -> 22	260	10	0	10	9
23 -> others	245	0	10	10	0
23 -> 23	245	9	1	10	8.1
24 -> others	235	0	10	10	0
24 -> 24	235	10	0	10	9
25 -> others	373	0	11	11	0
25 -> 25	373	5	6	11	4.091
26 -> others	155	0	10	10	0
26 -> 26	155	10	0	10	9
27 -> others	351	0	10	10	0
27 -> 27	351	8	2	10	7.2
28 -> others	480	2	8	10	1.8
28 -> 28	480	9	1	10	8.1
29 -> others	357	0	11	11	0
29 -> 29	357	11	0	11	9
30 -> others	355	0	10	10	0
30 -> 30	355	9	1	10	8.1
31 -> others	574	4	6	10	3.6
31 -> 31	574	9	1	10	8.1
32 -> others	349	3	7	10	2.7
32 -> 32	349	10	0	10	9
33 -> others	280	0	10	10	0
33 -> 33	280	10	0	10	9
34 -> others	275	0	10	10	0
34 -> 34	275	10	0	10	9
35 -> others	450	0	11	11	0
35 -> 35	450	11	0	11	9
36 -> others	362	0	14	14	0
36 -> 36	362	5	9	14	3.214
37 -> others	325	0	24	24	0
37 -> 37	325	24	0	24	9
38 -> others	165	0	11	11	0
38 -> 38	165	11	0	11	9
39 -> others	259	0	10	10	0
39 -> 39	259	10	0	10	9
40 -> others	316	0	10	10	0
40 -> 40	316	9	1	10	8.1
41 -> others	347	0	12	12	0
41 -> 41	347	3	9	12	2.25
42 -> others	298	0	9	9	0
42 -> 42	298	4	5	9	4
43 -> others	295	0	10	10	0
43 -> 43	295	10	0	10	9
44 -> others	637	0	10	10	0
44 -> 44	637	9	1	10	8.1

45 -> others	280	0	9	9	0
45 -> 45	280	9	0	9	9
46 -> others	335	0	11	11	0
46 -> 46	335	9	2	11	7.36
47 -> others	275	0	10	10	0
47 -> 47	275	4	6	10	3.6
48-> others	361	3	7	10	2.7
48 -> 48	361	8	2	10	7.2

Table 4.4.1 Confusion matrix for the analysis of the IOGear input device.

Signature	**Cutoff(85%)**	**No. Accepted**	**No. Rejected**	**Total**	**(TC/total)*min**
1 -> others	522	0	10	10	0
1 -> 1	522	3	7	10	2.7
2->others	448	0	10	10	0
2-> 2	448	6	4	10	5.4
3 -> others	684	4	6	10	3.6
3 -> 3	684	9	1	10	8.1
4 -> others	506	0	10	10	0
4 -> 4	506	9	1	10	8.1
5 -> others	505	0	10	10	0
5 -> 5	505	8	2	10	7.2
6 -> others	500	0	10	10	0
6 -> 6	500	10	0	10	9
7 -> others	450	0	10	10	0
7 -> 7	450	10	0	10	9
8 -> others	590	0	10	10	0
8 -> 8	590	10	0	10	9
9 -> others	513	0	10	10	0
9 -> 9	513	10	0	10	9
10 -> others	470	0	10	10	0
10 -> 10	470	9	1	10	8.1
11 -> others	700	0	10	10	0
11 -> 11	700	9	1	10	8.1
12 -> others	438	0	10	10	0
12 -> 12	438	10	0	10	9
13 -> others	698	4	6	10	3.6
13 -> 13	698	10	0	10	9
14 -> others	627	0	10	10	0
14 -> 14	627	5	5	10	4.5
15 -> others	716	2	8	10	1.8
15 -> 15	716	7	3	10	6.3
16 -> others	468	1	9	10	0.9
16 -> 16	468	10	0	10	9

17 -> others	3180	3	7	10	2.7
17 -> 17	3180	7	3	10	6.3
18 -> others	470	0	10	10	0
18 -> 18	470	10	0	10	9
19 -> others	546	0	10	10	0
19 -> 19	546	5	5	10	4.5
20 -> others	538	0	10	10	0
20->20	538	7	3	10	6.3
21->others	514	0	10	10	0
21->21	514	10	0	10	9
22 -> others	490	0	10	10	0
22 -> 22	490	10	0	10	9
23 -> others	440	0	10	10	0
23 -> 23	440	9	1	10	8.1
24 -> others	442	0	10	10	0
24 -> 24	442	9	1	10	8.1
25 -> others	4136	0	10	10	0
25 -> 25	4136	10	0	10	9
26 -> others	480	0	10	10	0
26 -> 26	480	10	0	10	9
27 -> others	1582	1	9	10	0.9
27 -> 27	1582	8	2	10	7.2
28 -> others	520	0	10	10	0
28 -> 28	520	10	0	10	9
29 -> others	612	0	10	10	0
29 -> 29	612	10	0	10	9
30 -> others	510	0	10	10	0
30 -> 30	510	9	1	10	8.1
31 -> others	1490	0	10	10	0
31 -> 31	1490	6	4	10	5.4
32 -> others	1098	0	10	10	0
32 -> 32	1098	6	4	10	5.4
33 -> others	573	0	10	10	0
33 -> 33	573	10	0	10	9
34 -> others	816	0	10	10	0
34 -> 34	816	9	1	10	8.1
35 -> others	460	0	10	10	0
35 -> 35	460	9	1	10	8.1
36 -> others	508	0	10	10	0
36 -> 36	508	10	0	10	9
37 -> others	595	0	10	10	0
37 -> 37	595	10	0	10	9
38 -> others	490	0	10	10	0
38 -> 38	490	10	0	10	9
39 -> others	495	0	10	10	0

39 -> 39	495	10	0	10	9
40 -> others	401	0	10	10	0
40 -> 40	401	10	0	10	9
41 -> others	519	0	10	10	0
41 -> 41	519	9	1	10	8.1
42 -> others	552	0	10	10	0
42 -> 42	552	9	1	10	8.1
43 -> others	532	0	10	10	0
43 -> 43	532	10	0	10	9
44 -> others	2462	1	9	10	0.9
44 -> 44	2462	9	1	10	8.1
45 -> others	553	0	10	10	0
45 -> 45	553	9	1	10	8.1
46 -> others	776	0	10	10	0
46 -> 46	776	10	0	10	9
47 -> others	593	1	9	10	0.9
47 -> 47	593	9	1	10	8.1
48-> others	410	0	10	10	0
48 -> 48	410	9	1	10	8.1

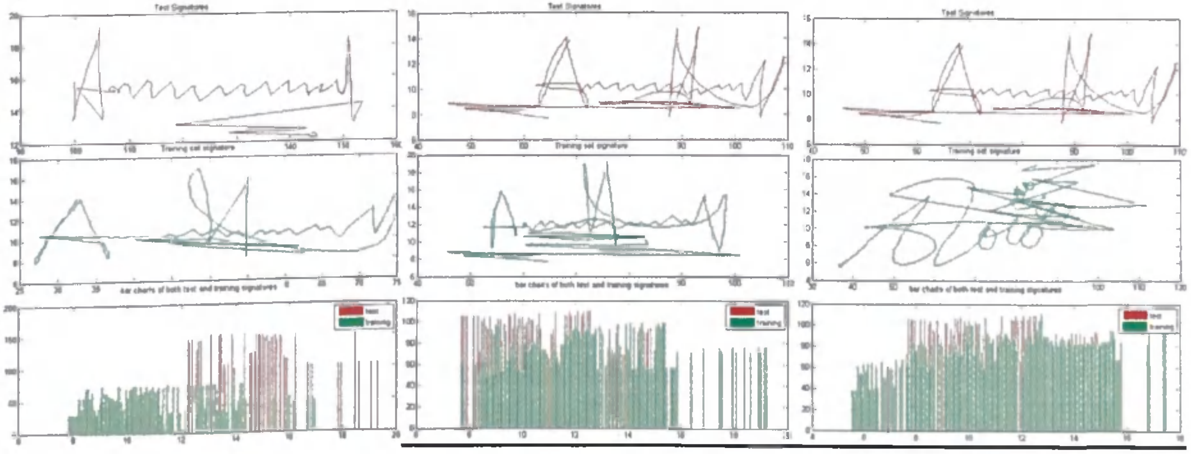
Table 4.4.2 Confusion matrix for the analysis of the I-pen input device.

Signature	Cutoff (85%)	No. Accepted	No. Rejected	Total	(TC/total)*min
1 -> others	1253	2	8	10	2
1 -> 1	1253	8	2	10	8
2->others	371	0	10	10	0
2-> 2	371	10	0	10	10
3 -> others	371	0	10	10	0
3 -> 3	371	5	5	10	5
4 -> others	380	0	10	10	0
4 -> 4	380	9	1	10	9
5 -> others	422	0	10	10	0
5 -> 5	422	10	0	10	10
6 -> others	364	0	10	10	0
6 -> 6	364	10	0	10	10
7 -> others	324	0	10	10	0
7 -> 7	324	10	0	10	10
8 -> others	463	0	10	10	0
8 -> 8	463	10	0	10	10
9 -> others	386	0	10	10	0
9 -> 9	386	9	1	10	9
10 -> others	419	0	10	10	0
10 -> 10	419	9	1	10	9
11 -> others	403	0	10	10	0
11 -> 11	403	10	10	10	10

12 -> others	386	0	10	10	0
12 -> 12	386	10	0	10	10
13 -> others	728	0	10	10	0
13 -> 13	728	5	5	10	5
14 -> others	708	0	10	10	0
14 -> 14	708	7	3	10	7
15 -> others	893	2	8	10	2
15 -> 15	893	8	2	10	8
16 -> others	422	0	10	10	0
16 -> 16	422	9	1	10	9
17 -> others	1837	1	9	10	1
17 -> 17	1837	6	4	10	6
18 -> others	455	0	10	10	0
18 -> 18	455	8	2	10	8
19 -> others	532	1	9	10	1
19 -> 19	532	6	4	10	6
20 -> others	470	0	10	10	0
20->20	470	3	7	10	3
21->others	412	0	10	10	0
21->21	412	8	2	10	8
22 -> others	568	2	8	10	2
22 -> 22	568	6	4	10	6
23 -> others	447	0	10	10	0
23 -> 23	447	10	0	10	10
24 -> others	380	0	10	10	0
24 -> 24	380	4	6	10	4
25 -> others	412	0	10	10	0
25 -> 25	412	10	0	10	10
26 -> others	316	0	10	10	0
26 -> 26	316	4	6	10	4
27 -> others	585	0	10	10	0
27 -> 27	585	8	2	10	8
28 -> others	400	0	14	14	0
28 -> 28	400	14	0	14	10
29 -> others	442	0	10	10	0
29 -> 29	442	10	0	10	10
30 -> others	590	3	7	10	3
30 -> 30	590	10	0	10	10
31 -> others	932	3	7	10	3
31 -> 31	932	8	2	10	8
32 -> others	486	4	10	14	2.857
32 -> 32	486	6	8	14	4.286
33 -> others	400	0	10	10	0
33 -> 33	400	10	0	10	10
34 -> others	496	0	10	10	0

34 -> 34	496	6	4	10	6
35 -> others	444	3	7	10	3
35 -> 35	444	8	2	10	8
36 -> others	452	4	6	10	4
36 -> 36	452	9	1	10	9
37 -> others	450	0	10	10	0
37 -> 37	450	10	0	10	10
38 -> others	363	1	9	10	1
38 -> 38	363	10	0	10	10
39 -> others	420	0	10	10	0
39 -> 39	420	10	0	10	10
40 -> others	310	0	10	10	0
40 -> 40	310	10	0	10	10
41 -> others	420	0	10	10	0
41 -> 41	420	7	3	10	7
42 -> others	1236	0	10	10	0
42 -> 42	1236	8	2	10	8
43 -> others	370	0	10	10	0
43 -> 43	370	10	0	10	10
44 -> others	900	0	10	10	0
44 -> 44	900	5	5	10	5
45 -> others	340	0	10	10	0
45 -> 45	340	10	0	10	10
46 -> others	400	0	10	10	0
46 -> 46	400	9	1	10	9
47 -> others	500	1	9	10	1
47 -> 47	500	7	3	10	7
48-> others	410	0	10	10	0
48 -> 48	410	10	0	10	10

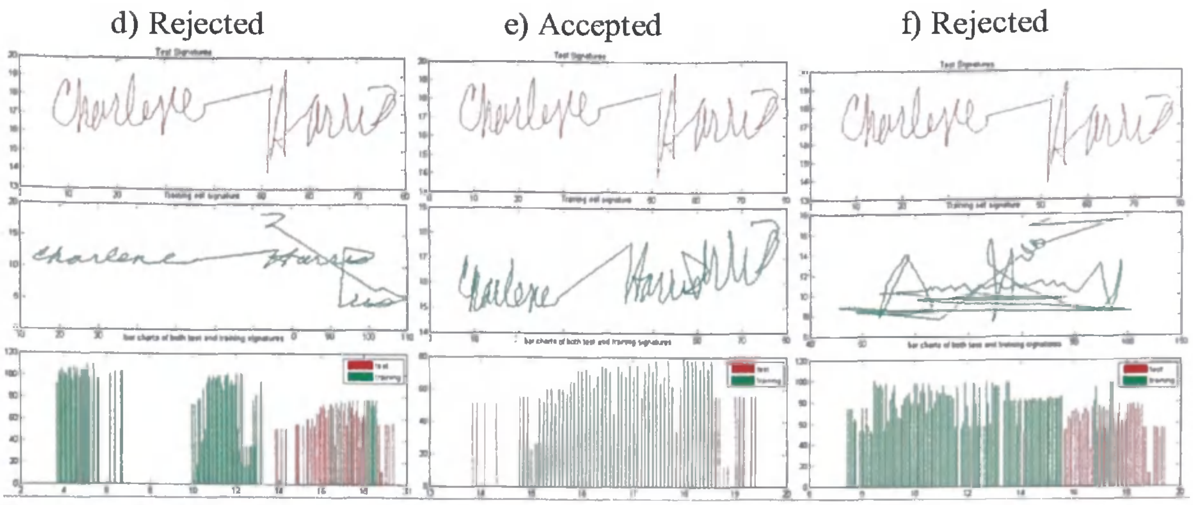
Table 4.4.3 Confusion matrix for the analysis of the mouse input device.



a) Rejected

b) Accepted

c) Rejected

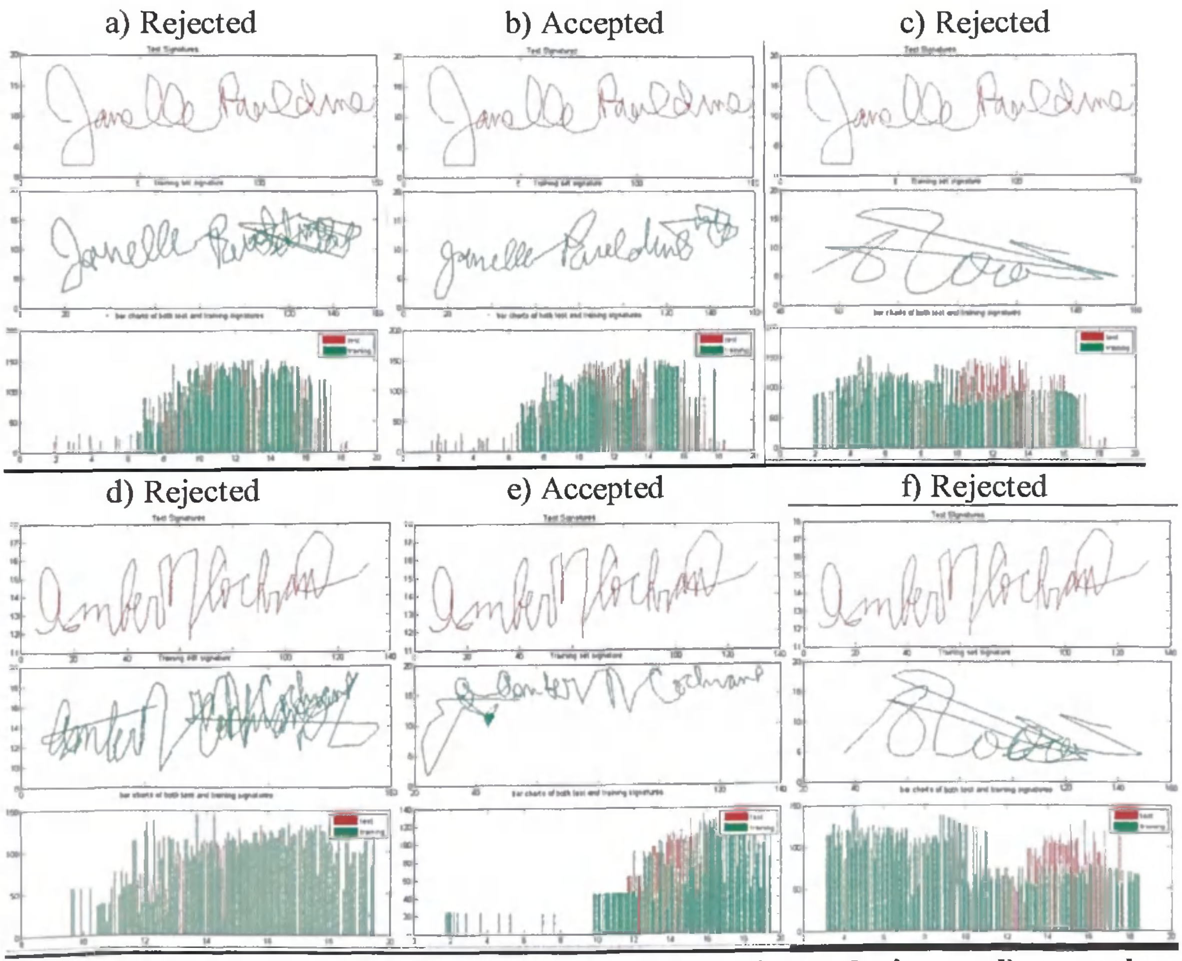


d) Rejected

e) Accepted

f) Rejected

Figure 4.4.1 Classification of signature from IOGear input device. a, d) examples of rejected signatures after individual to individual classification. b, e) examples of accepted signatures after individual to individual classification. c, f) examples of rejected signatures after individual to others classification.



a) Rejected

b) Accepted

c) Rejected

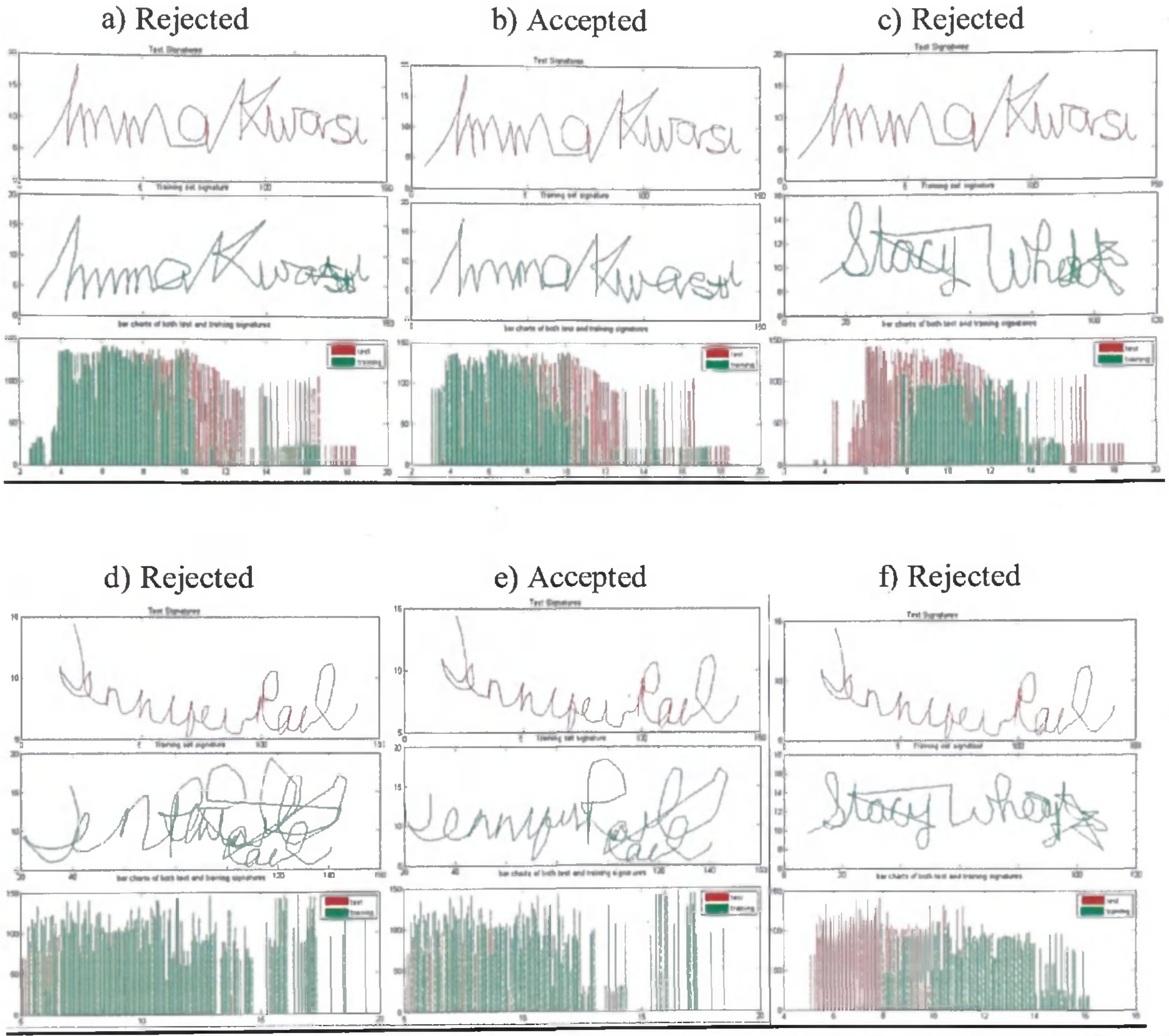
d) Rejected

e) Accepted

f) Rejected

Figure 4.4.2 Classification of signature from I-Pen input device. a, d) examples of rejected signatures after individual to individual classification. b, e) examples of

accepted signatures after individual to individual classification. c, f) examples of rejected signatures after individual to others classification.


a) Rejected

b) Accepted

c) Rejected

d) Rejected

e) Accepted

f) Rejected

Figure 4.4.3 Classification of signature from mouse input device. a, d) examples of rejected signatures after individual to individual classification. b, e) examples of accepted signatures after individual to individual classification. c, f) examples of rejected signatures after individual to others classification.

Input device	True Correct (TC)	False Correct (FC)	True Incorrect (TI)	False Incorrect (FI)	Precision (P)
IOGear	351.993/432	14.55/432	417.45/432	80.01/432	351.993/(14.55+351.993)
I-pen	380.7/432	15.3/432	416.7/432	51.3/432	380.7/(15.3+380.7)
mouse	393.286/490	25.857/490	454.143/490	96.714/490	393.286/(25.857+393.286)

Table 4.4.4 Performance evaluation for the three input devices.

Input device	Probability of classification as correct	Probability of classification as incorrect
IOGear (individual)	0.8148	0.1852
IOGear (others)	0.0337	0.9663

Table 4.4.5 Classification confusion matrix for IOGear input device

Input device	Probability of classification as correct	Probability of classification as incorrect
I-Pen (individual)	0.8813	0.1188
I-Pen (others)	0.0354	0.9646

Table 4.4.6 Classification confusion matrix for I-Pen input device

Input device	Probability of classification as correct	Probability of classification as incorrect
I-Pen (individual)	0.8026	0.1974
I-Pen (others)	0.0528	0.9268

Table 4.4.7 Classification confusion matrix for mouse input device

The correct and incorrect classification tests were conducted for all the three input devices. To avoid the possibility of the results being biased, the percentage correct was computed for each signature and the values were added which resulted in the values in Table 4.4.4 above. It was observed that for an individual to individual classification, the IOGear input device reported 0.8148 as the probability of true correct classification which is approximately 81% where as the probability of true incorrect classification was 0.1852 which is also approximately 19%. Individual to other people's classification for the same input device was recorded as 0.0337 for probability of false correct classification which is approximately 3.4% where as the probability of false incorrect classification was 0.9663, approximately 96.6%.

For the I-pen input device, individual to individual classification resulted in 0.8813 as the probability of true correct classification which is approximately 88% where as the probability of true incorrect classification was 0.1188 which is also approximately 12%. Like IOGear input device, individual to other people's classification resulted in 0.0354 as the probability of false correct classification which is approximately 3.5% where as the probability of false incorrect classification recorded 0.9646, approximately 96%.

Finally, individual to individual classification for the mouse input device recorded 0.8026 as the probability of true correct classification which is approximately 80% where the probability of true incorrect classification was 0.1974 which is also approximately 20%. For an individual to other people's classification for the mouse input device, 0.0528 was recorded as the probability of false correct classification which is approximately was recorded as the probability of false correct classification which is approximately 5.3% where as the probability of false incorrect classification was 0.9268, approximately 92.7%. These classifications are displayed in Table 4.4.5, Table 4.4.6 and Table 4.4.7.

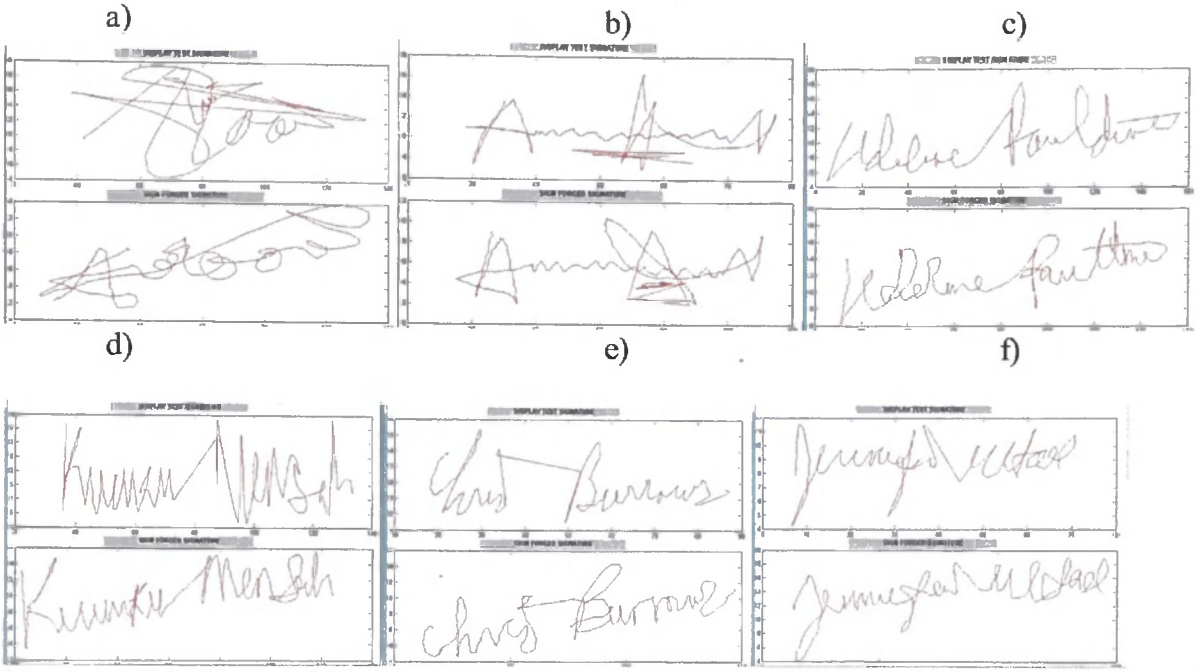


Figure 4.4.4 Single signature analysis for IOGear input device a,b,c) Examples of rejected signatures d,e,f) Examples of accepted signatures

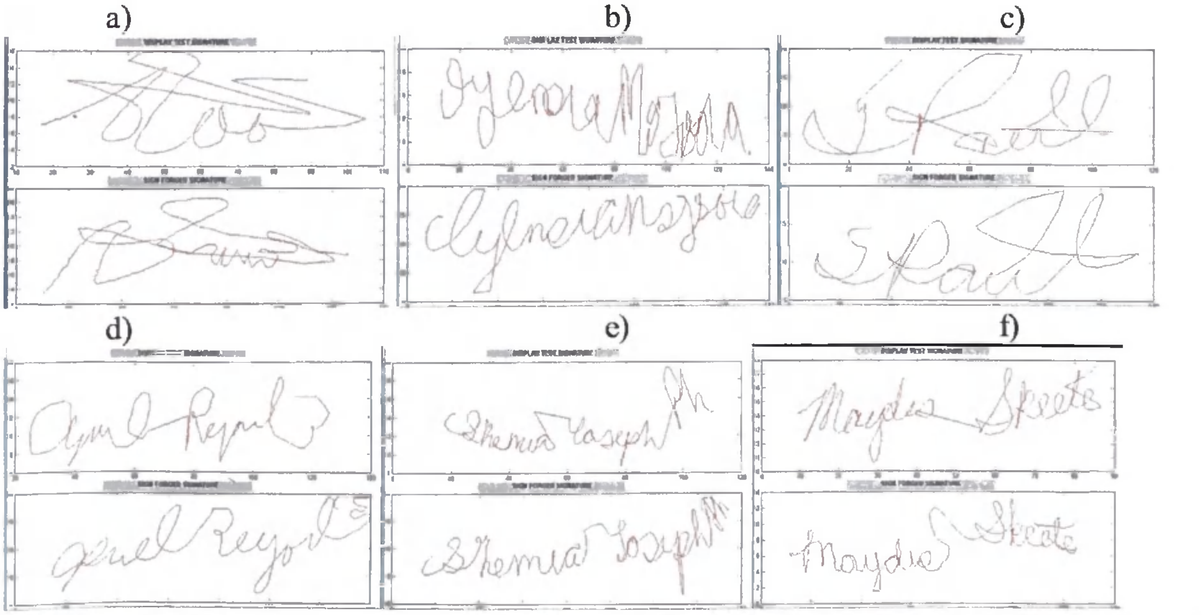


Figure 4.4.5 Single signature analysis for I-Pen input device a,b,c) Examples of rejected signatures d,e,f) Examples of accepted signatures

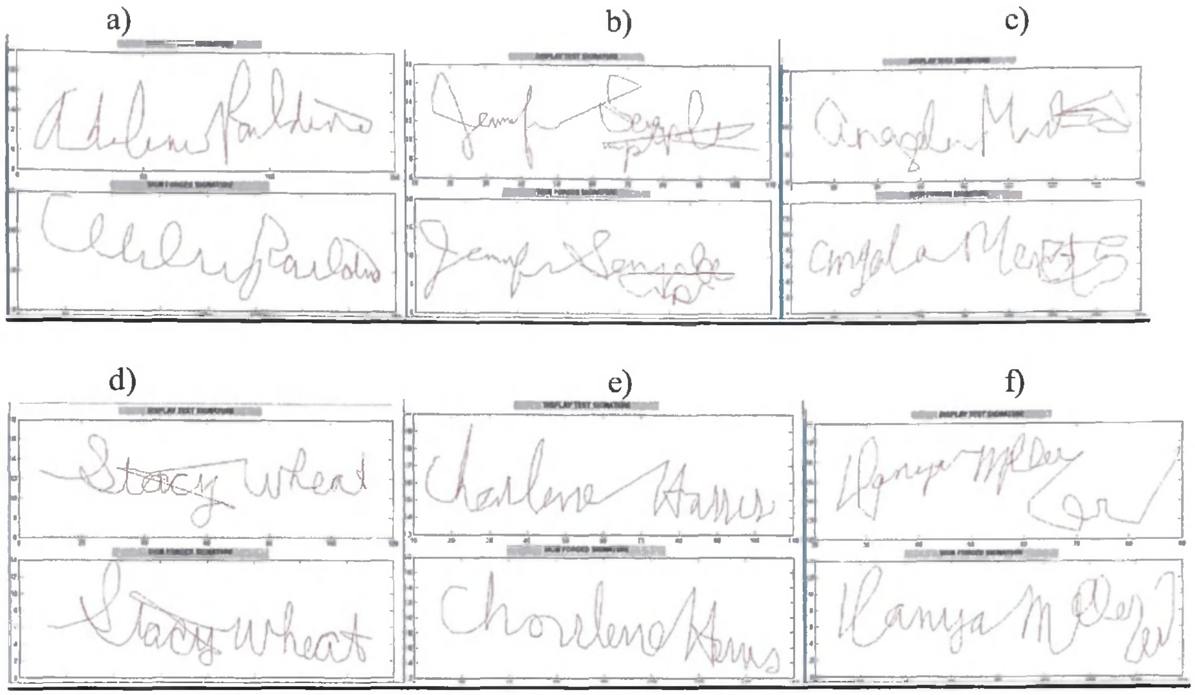


Figure 4.4.6 Single signature analysis for mouse input device a,b,c) Examples of rejected signatures d,e,f) Examples of accepted signatures

Examples of correct and incorrect classification using the fraud detection tool for the three selected input devices are provided in Figure 4.4.4, Figure 4.4.5 and Figure 4.4.6 above. The results displayed in Table 4.4.2 below, shows clearly the rate at which people's signature could be forged using the selected input devices. After the experiment, it was observed that, the I-pen input device stands the higher risk of people forging other people's signature (55%). The fraud rate for both IOGear and the mouse were promising as compared to the I-pen (25% and 35% respectively). This could be attributed to the fact that, other special security features such as speed, pressure between the pen and the paper, the angel at which the pen was tilt etcetera, were not collected during the forging of the signature, the forged signature was signed with a considerable amount of time.

	IOGear	I-Pen	Mouse
Accept rate (correct/total)	5/20=0.25	11/20=0.55	7/20=0.35
Reject rate (incorrect/total)	15/20=0.75	9/20=0.45	13/20=0.65
Fraud rate	0.25*100=25%	0.55*100=55%	0.35*100=35%

Table 4.4.8 the rate of fraud for each input device

During the capturing of the various signatures, the users who signed their names were asked two questions. The first question was "Which input device do you like best?" and the second question was "If the first choice is not available which one will you choose?" These answers were collected and entered into an SPSS and the quantitative analysis was run on those data. From the quantitative analysis of the three input devices it was deduced that out of the 59 users 13 were missing data which was 22% of the total as illustrated in Table 4.4.9 below. Table 4.4.9 also recorded 15.2%, 26.1% and 58.7% as a valid percent of users who selected the mouse as their first choice, second choice and third choice input devices respectively. Table 4.4.10, recorded 6.5%, 63% and 30.4% as a valid percentage of users who selected the I-Pen as their first choice, second choice and third choice respectively. Finally, Table 4.4.11, recorded 73.3%, 10.9% and 10.9% as a valid percentage of users who selected the I-Pen as their first choice, second choice and third choice as their favorite input device respectively. The bar charts of these results are illustrated in Figure 4.4.7.

Mouse

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	first choice	7	11.9	15.2	15.2
	second choice	12	20.3	26.1	41.3
	third choice	27	45.8	58.7	100.0
	Total	46	78.0	100.0	
Missing	998	13	22.0		
Total		59	100.0		

Table 4.4.9 the quantitative analysis of the Mouse input device

I-Pen

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	first choice	3	5.1	6.5	6.5
	second choice	29	49.2	63.0	69.6
	third choice	14	23.7	30.4	100.0
	Total	46	78.0	100.0	
Missing	998	13	22.0		
Total		59	100.0		

Table 4.4.10 the quantitative analysis of the I-Pen input device

IOGear

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	first choice	36	61.0	78.3	78.3
	second choice	5	8.5	10.9	89.1
	third choice	5	8.5	10.9	100.0
	Total	46	78.0	100.0	
Missing	998	13	22.0		
Total		59	100.0		

Table 4.4.11 the quantitative analysis of the IOGear input device

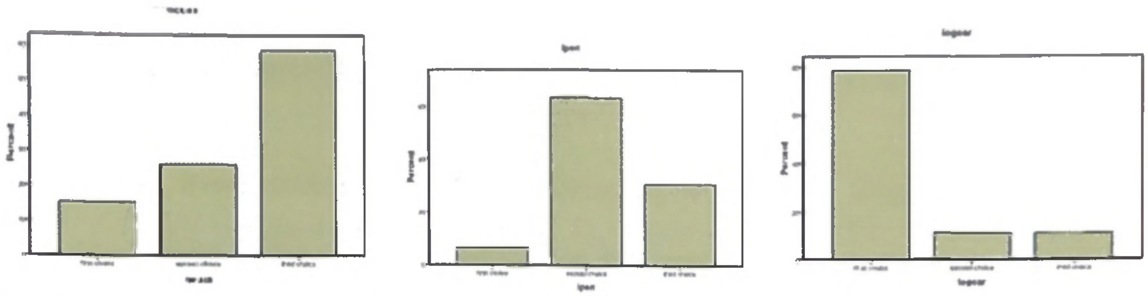


Figure 4.4.7 a bar chart showing the percentage of users' first choice, second choice and third choice for the respective input device (a) A bar chart for the mouse input device (b) A bar chart for the I-Pen input device (c) A bar chart for the IOGear input device.

4.5 RECOMMENDATION

Since this research is part of a broader scope, it is of higher recommendation that this research work be extended to cover the string analysis of the signature as described in details in section two of this research work. Leclerc and Plamondon's [28] six steps approach for processing handwritten signature could be implemented for better performance. I recommend practices such as smoothing, filtering, wild point correction, dehooking, dot direction and stroke connection before and after signature capture to control the level of noise. I recommend that is research work be extended by combining both feature comparison and temporal functions in signature verification process. There is a higher possibility of achieving better performance when both are implemented within a system. In addition to this, parametric classification could be implemented as compared to non-parametric technique used in this research paper.

interest among the three input devices was the IOGear but as to the sensitivity of flexibility of implementing one input device could be an interesting research extension.

The fraud detection tool was designed to test the forging of signatures on individual bases. That is after determining the cutoff of that individual, that cutoff the point of decision as to whether the forged signature is to be accepted or rejected. I recommend that this tool be extended to implement a more secured by determining the speed at which the forger is signing the signature. This will bring some sort of robustness into the fraud detection process.

After taking a critical look at the analysis of the selected input devices there is one specific input device that stands out based on user's interest, flexibility of use, easier adaptation and the most accepted input device by users. From the statistical point of view based on users interest this research paper recommend the use of the IOGear as the best input device well accepted by the population that signed their names for the purpose of this research work.

5. CONCLUSIONS AND FUTURE WORK

This section is the conclusion and the projection of the future works that could be implemented to extend this idea. The conclusion will outline the summary of the research work, the successes and the failures of this research work. The Future work will be focused on projecting other alternatives and other methodologies that could be implemented to advance this research idea as technology keeps on changing every now and then and different ideas keeps popping up.

5.1 CONCLUSION

The primary objective of this research paper was to develop a principal approach to solving the problems associated with file encryption, computer access, and data protection using password and other biometric means. In spite of the increased knowledge of protecting data and unauthorized computer access by creating complex password and frequently changing passwords, there has been limited research exploring the possibility of using handwritten signatures to protect data and unauthorized computer access. However, the problem of replacing the current system of data and file protections with more secured, flexible and user friendly system has been very challenging. The aim of this research paper was to explore this option and to overcome its challenges. The objective was not limited to finding a solution to the problem but it also included development of an application that will help explain better the new technology proposed. Another idea which supplement the problem was to seek the user's view as which among selected input devices they will feel more flexible and easily adapted to.

The methodologies implemented in this research work provide higher classification accuracy. The fuzzy decision rules and the membership functions which shows the overlapping classes gave the best fuzzy classification of the signatures. The research paper came to a conclusion that fuzzy classification of handwritten signatures is an accurate and efficient way of granting users access to their computers and also encrypting their documents as compared with various kinds of signatures and various biometric means of protecting data from unauthorized users. In general a good replacement for complex password idea, frequent change of password, becoming a diligent user in order to protect your documents, keeping and remembering password is the fuzzy classification of handwritten signature. This research work demonstrated the how secured users will be when their computers and documents are been protected by handwritten signatures. It was observed that after running the individual signature against other people's signature 100% of the time it was rejected. That is, when someone tries to access your computer or document using the handwritten signature he/she made he/she will be rejected.

From the quantitative analysis from sub-section 4.3 it could be concluded that the input device most users preferred as their first choice was IOGear. The IOGear input device was recorded as 78.3% as compared to the 15.2% and 6.5% of the mouse and the I-Pen respectively. Alternatively, the I-pen was the next favorite input device for most of the users if the IOGear was not an option. It was recorded that 63% of users selected the I-Pen as their next favorite input device as compared to the 26.1% and 10.9% of the mouse and IOGear respectively. Finally, the last option (mouse) was the input device most users selected as their last favorite amongst the three input devices. It was recorded that, 58.7% of users selected the mouse as their third favorite as compared to 30.4% and 10.9% for I-Pen and IOGear respectively.

Several conclusions could be drawn from this outcome. Some of the conclusions that could be drawn out of this result are as follows; Firstly, since the IOGear input device is like an ordinary pen, users seems to feel more comfortable using them to sign their name on a piece of paper. Secondly, notebook computer do not have mouse attached to them as the desktop computers. Due to this reason computer users do not have frequent usage of this input device which makes it new to them. Thirdly, with the IOGear input device, you can sign your name without necessarily looking on the computer screen as compared to the mouse and the I-Pen input devices. Finally, it feels more luxurious and more technology wise to write on paper and see it document on the computer screen. Users had fun signing their names with the IOGear input device than the mouse and I-Pen.

5.2 FUTURE WORK

There are several ideas that could be implemented to extend the work presented in this research paper.

1. ***Input Device:*** One promising direction and a fair challenge is to try different input devices that captures more advanced features such as the pen tilt during the capturing of the signature, the pressure of the input device to the writing surface during the capturing of the signature and other features that will enhance the security aspect of this research work.
2. ***Classification:*** Another part of this research which was not accomplished in this research work due to time constraints was analyzing the collected signatures as a string. Most of the background work was done to cover the analysis of the signatures as a string but was left out for future expansion of this idea. The description, classification and recognition of handwritten data are most commonly focused on character by character recognition. There are different kinds of characters ranging from numeric characters to symbols. These characters come in different scripts, languages, sizes, and shapes. In the case of character classification and recognition, spacing of individual characters in a string of characters should be considered. String or text classification and recognition

overlook the spacing, shape and size of the individual characters and consider the string of characters together as one which makes recognition and classification much simpler as compared to the individual character recognition. Aspect normalization could be implemented to extend the scope of this research work as recommended by Nalwa [30], since most writers do not write their signature along the vertical and the horizontal dimensions. Most of the time, the same writer might write their signature bigger and shorter and different times make their signature taller and longer. In addition to this, instead of finding the threshold of individual signatures using the 85% confidence level, we could find the threshold that equalizes the probabilities of false accept and false rejection. This method is known to be a figure of merit of a verification system as stated by Plamondon [27].

3. ***Algorithm:*** In addition to the algorithms already discovered in this research work, the Frobenius Norm which is also called Euclidean Norm could also be implemented in expanding this idea. The Frobenius Norm could also be considered as a Vector Norm. The ultimate goal of handwritten signature classification is to be able to allow user to use the system without the fear of someone signing their signature to get access to their protected files and data. Another positive direction towards achieving this goal is defuzzification of the fuzzified signatures. Defuzzification is a procedure to calculate the single representative value of a fuzzy set. Either the center of gravity or the mean of maxima could be implemented.

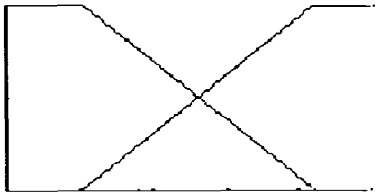
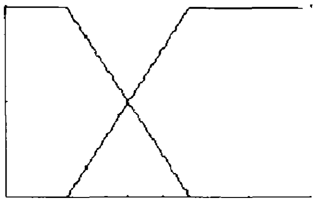
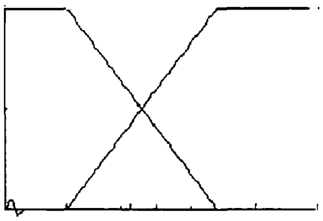
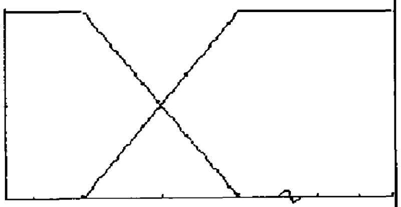
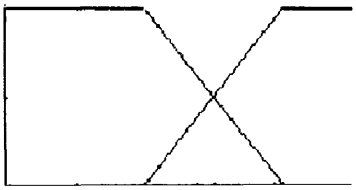
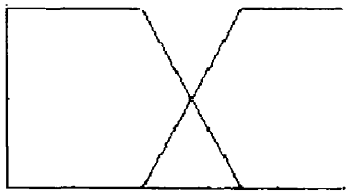
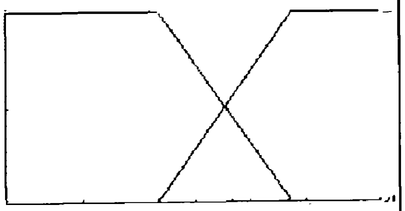
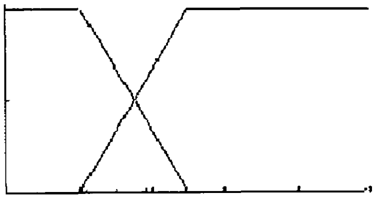
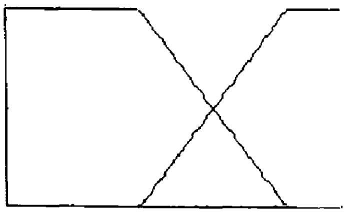
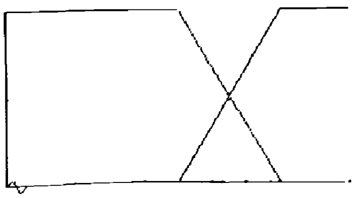
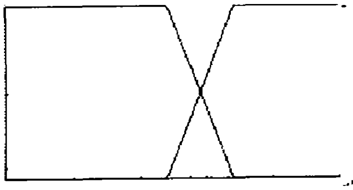
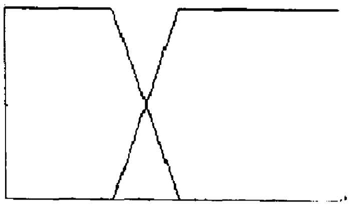
4. ***Classifiers:*** There are a lot more classifiers but what I recommend for future extension of this project are Mamdani-Assilian (MA) model and Takagi-Sugeno-Kang (TSK) model [84]. In MA systems both the input and the output are represented by linguistic terms. In TSK models the antecedent part of the rule is Boolean but the consequent is a function of the input. An in-depth documentation is presented by Ludmila [84].

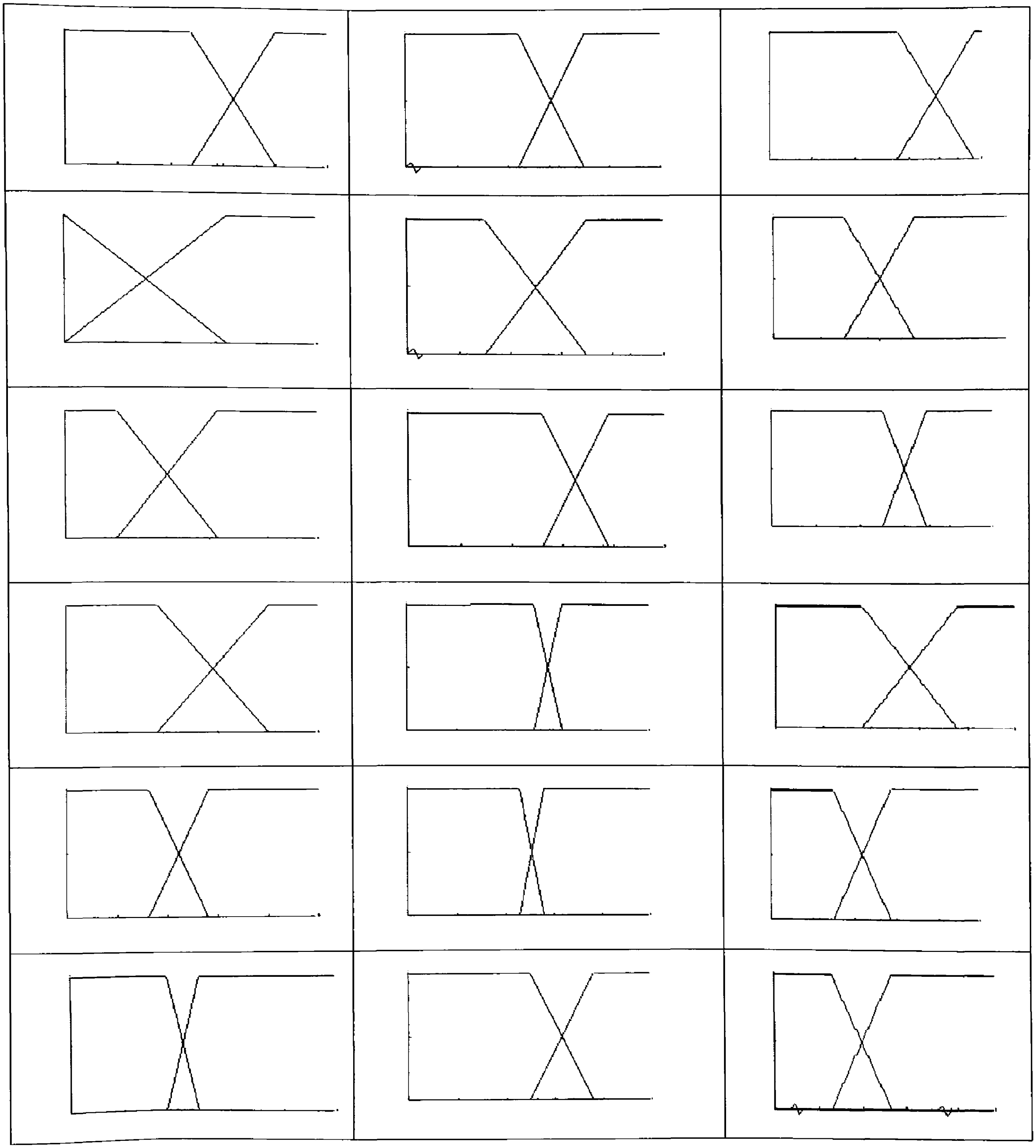
LIST OF FIGURES

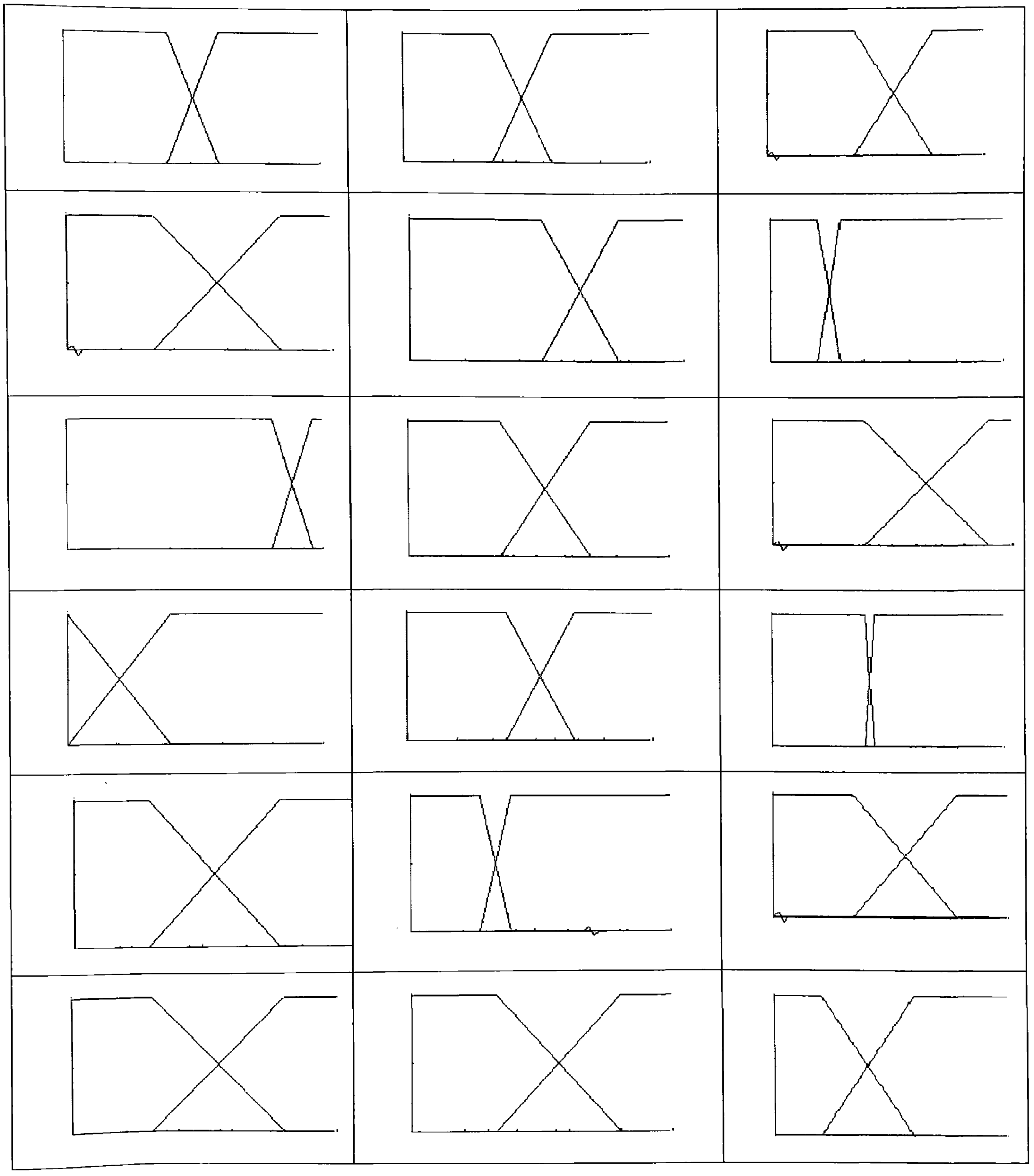
LIST OF TABLES

APPENDIX A

This section provides the tabulated membership functions for all the three input devices. Each cell in the tables below is a representation of the membership function generated for an individual's signatures signed with the respective input devices. Table A1, Table A2 and Table A3 are the membership functions generated for individuals using the IOGear, I-pen and the mouse input devices respectively.





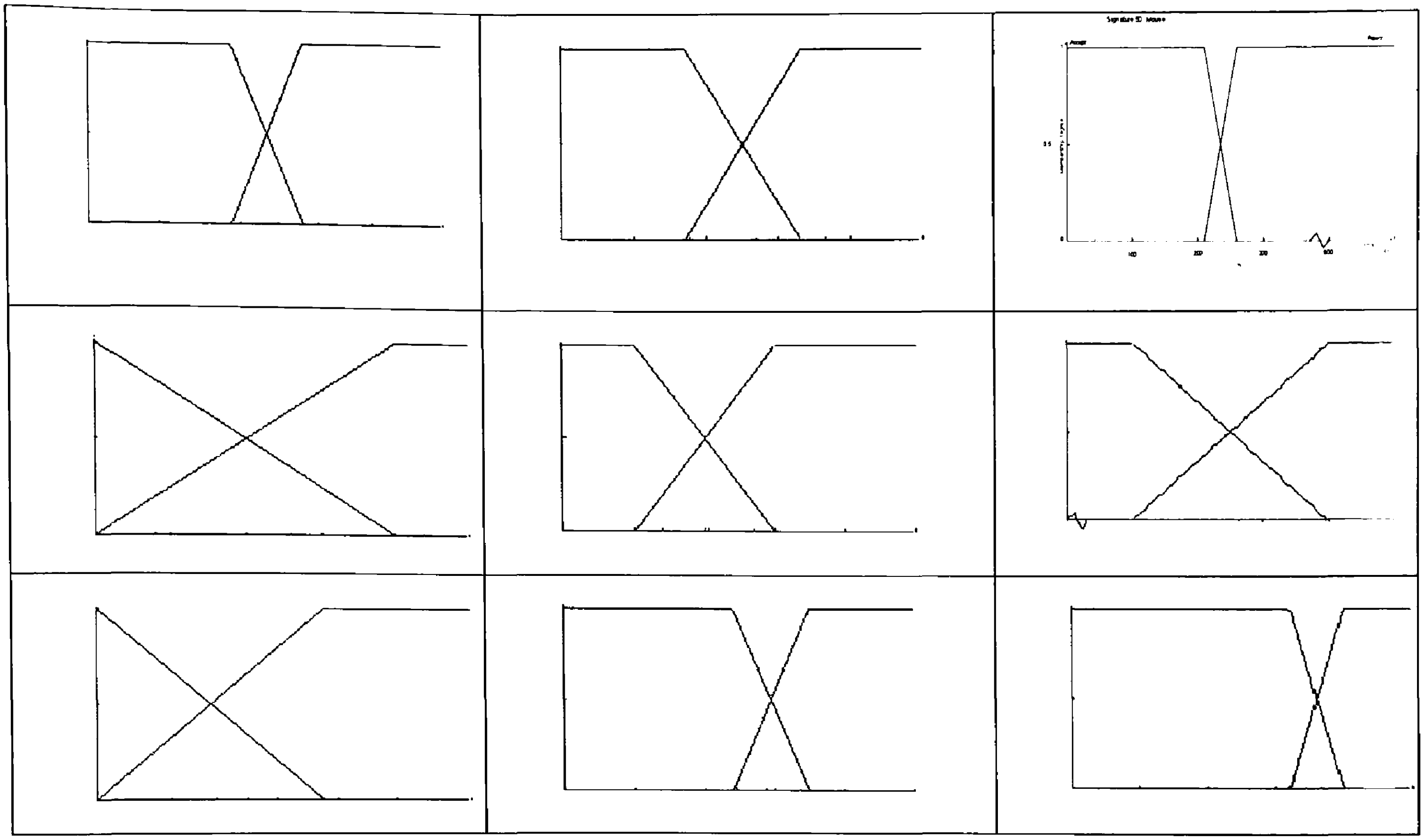
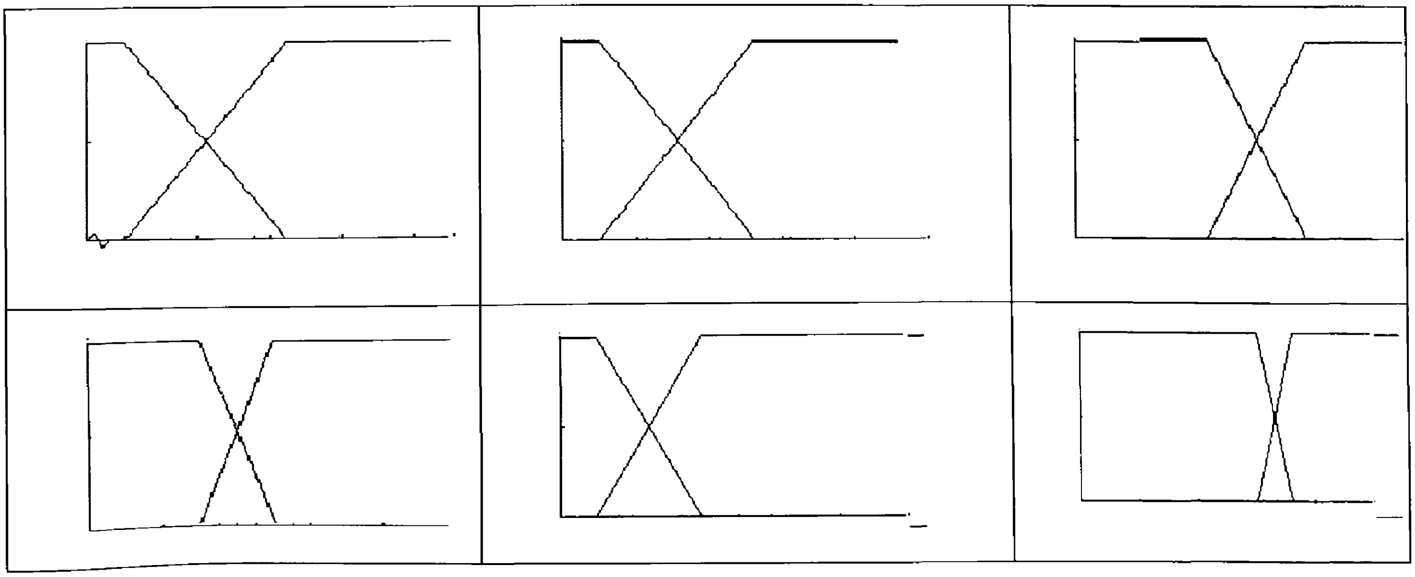
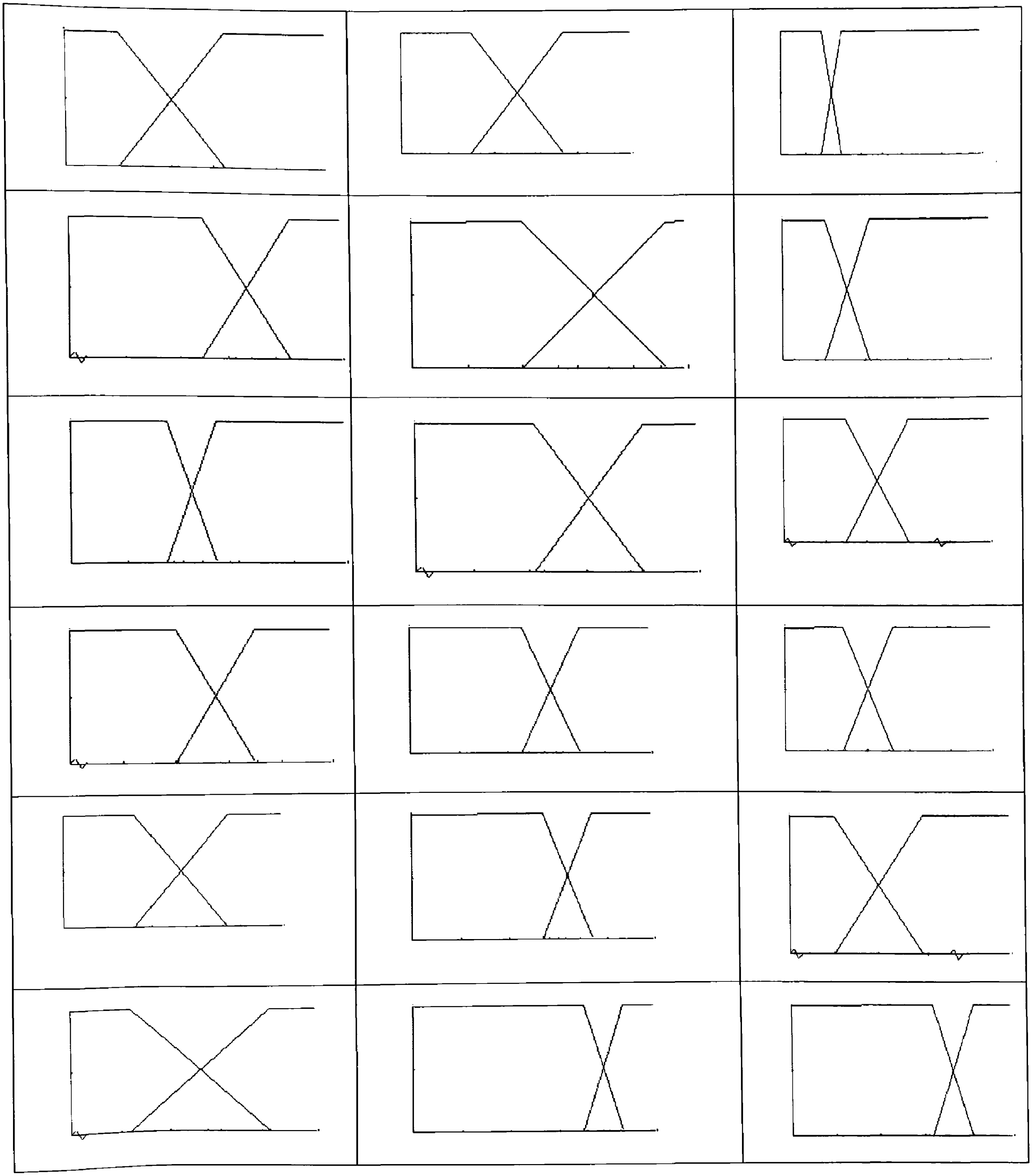
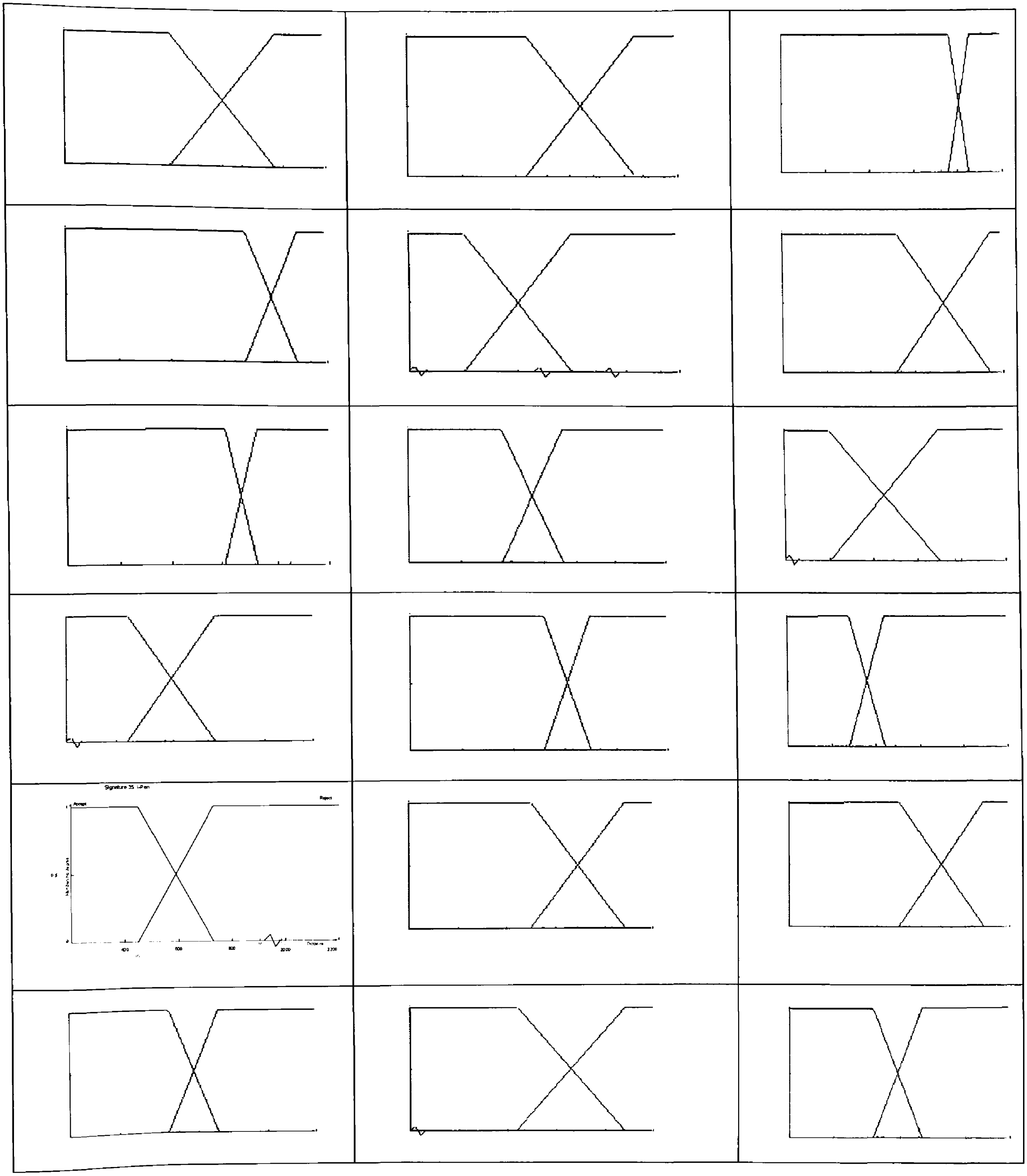


Table A.1 Membership functions of signatures using the IOGear input device






Signature 35 I-Pen
Accept
Reject
1
0.5
0
400
600
800
2000
2200
Distance

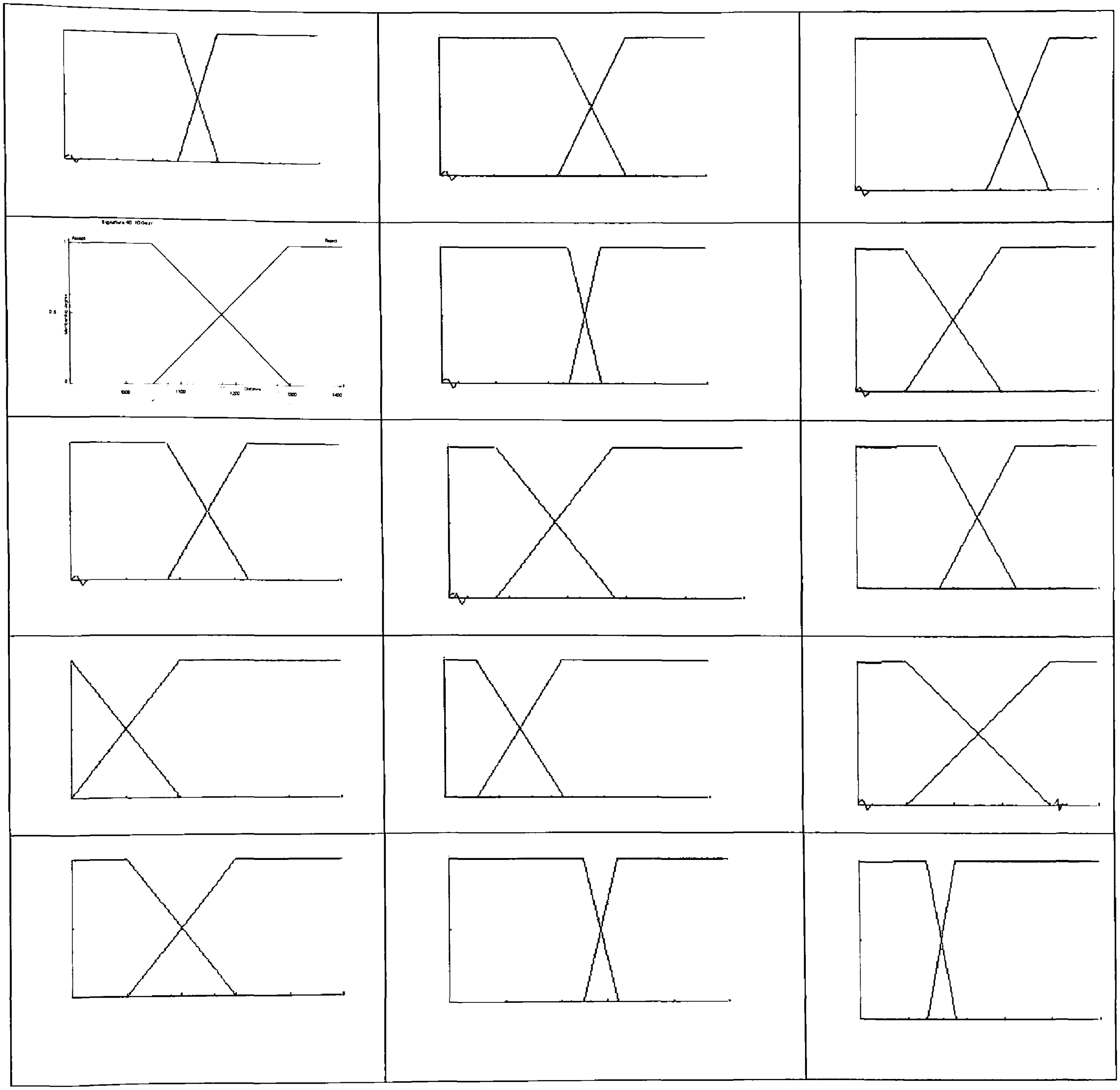
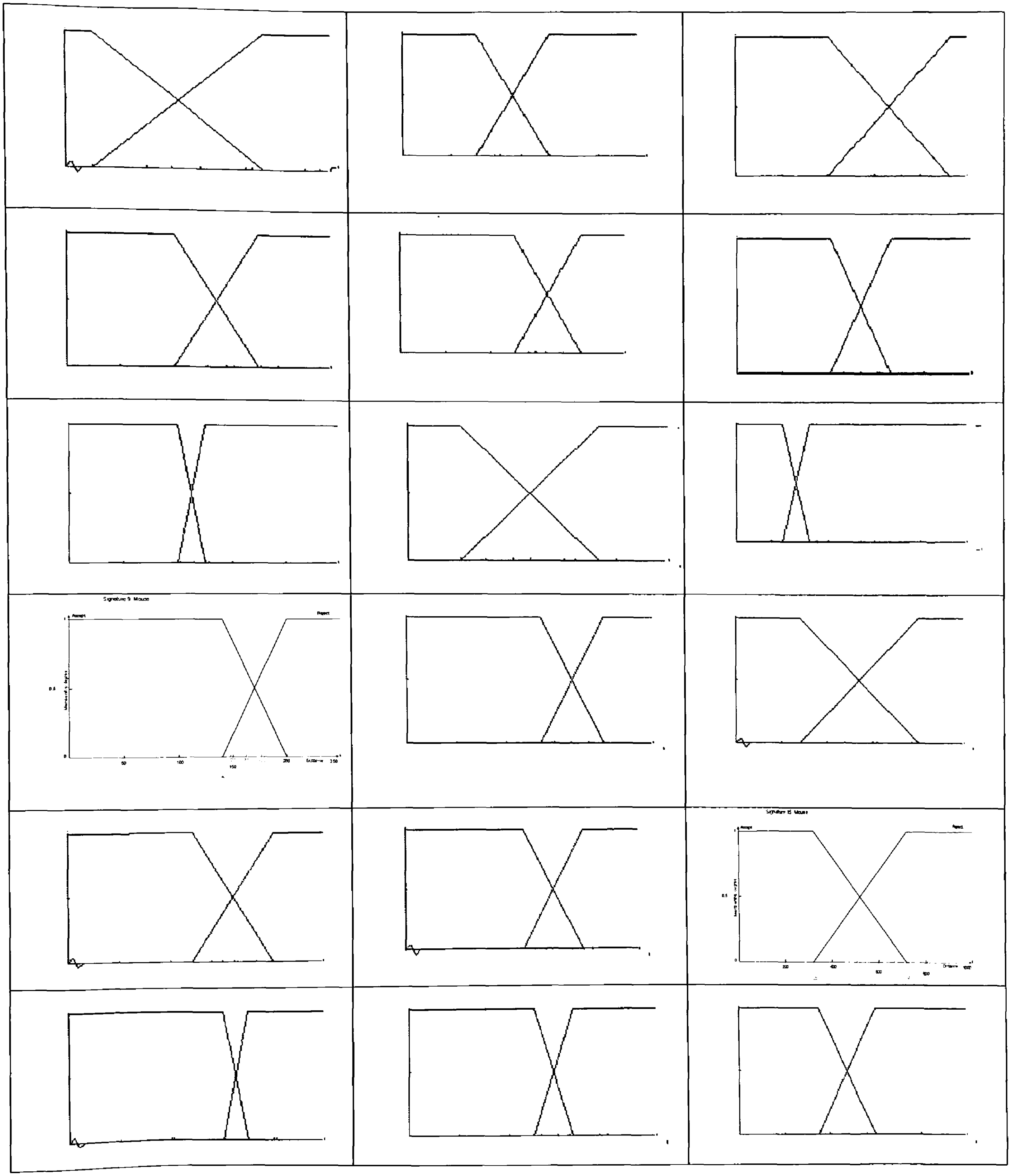
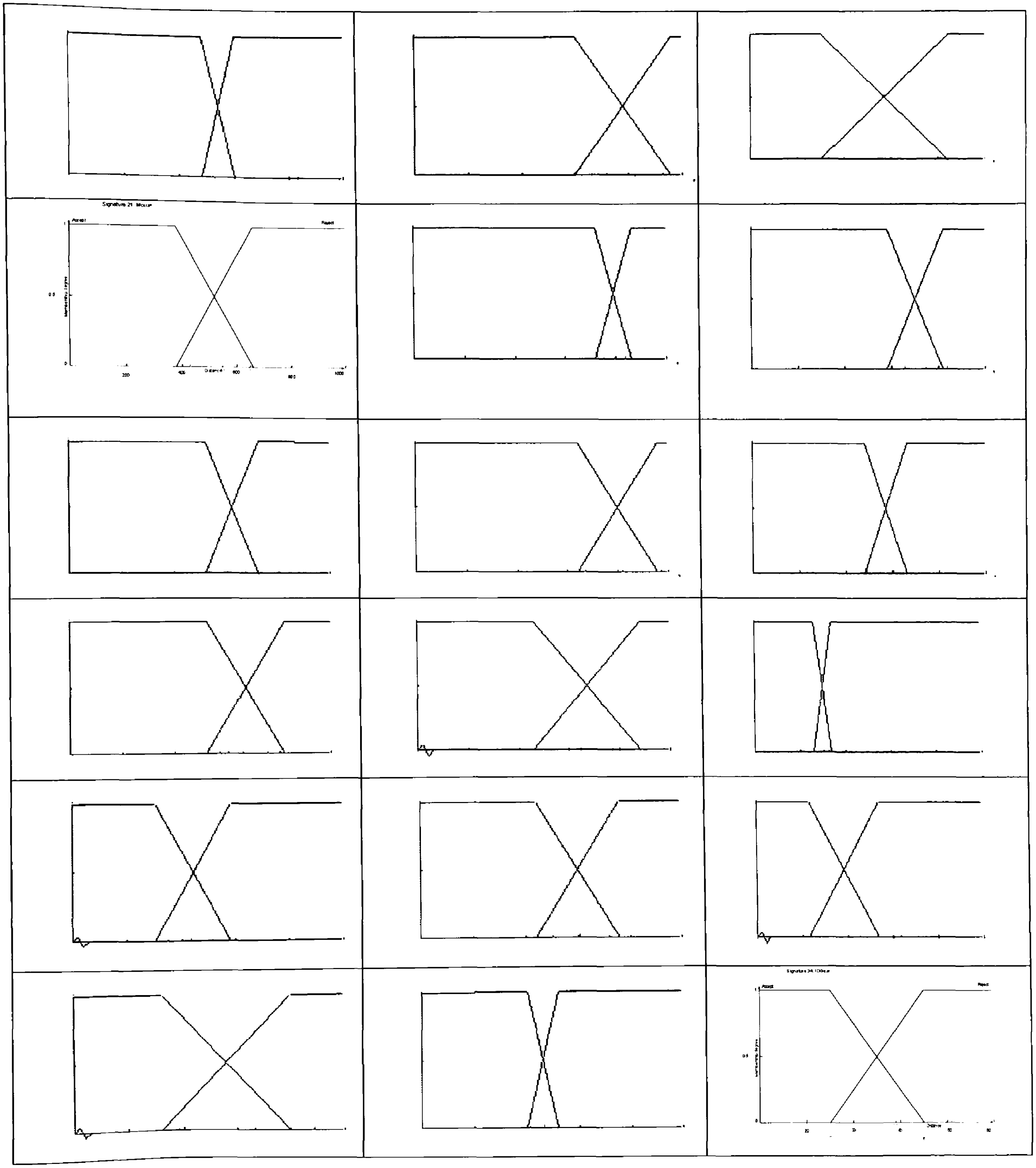
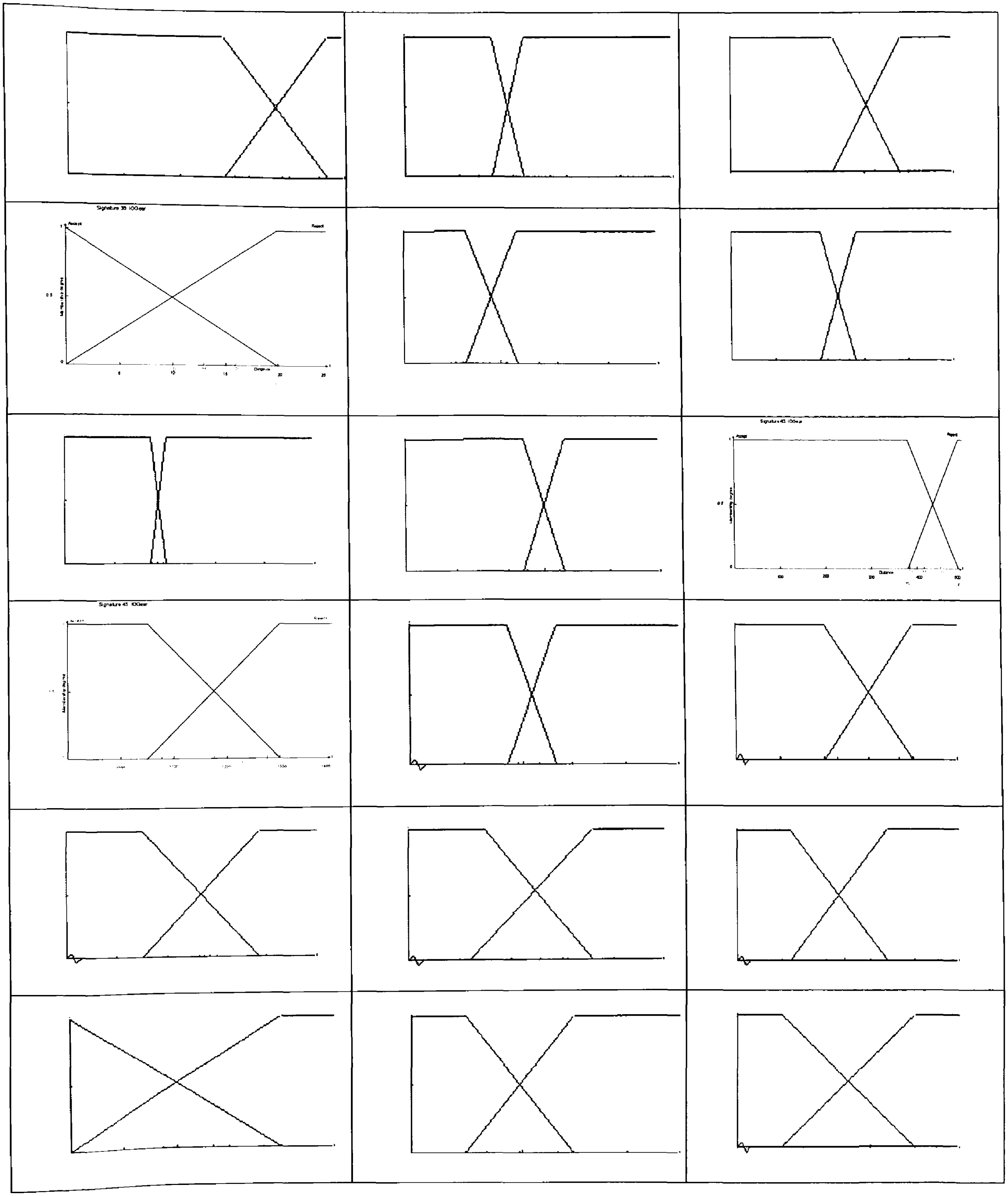


Table A.2 Membership functions of signatures using the I-Pen input device


Signature 9 Mouse
Accept
Reject
1
0.5
0
Membership degree
50
100
150
200
Distance
250
Signature 15 Mouse
Accept
Reject
1
0.5
0
Membership degree
200
400
600
800
Distance
1000


Signature 21 Mouse
Accept
Reject
1
0.5
Membership degree
0
200
400
Distance
600
800
1000
Signature 34 100eur
Accept
Reject
1
0.5
Membership degree
0
10
20
Distance
40
50
60


Signature 38 100ear
Accept
Reject
1
0.5
0
5
10
15
20
25
Degrees
Signature 43 100ear
Accept
Reject
1
0.5
0
Membership degree
100
200
300
400
500
Distance
Signature 45 100ear

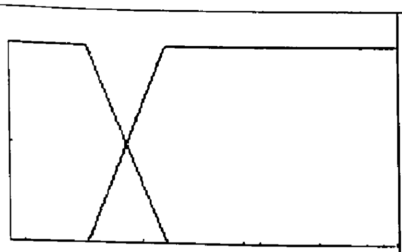
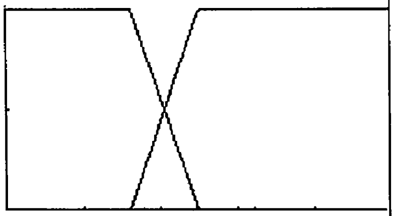
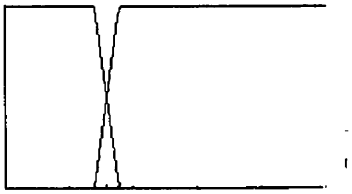
		

Table A.3 Membership functions of signatures using the Mouse input device

REFERENCES

[1] C. C. Lin and Chellappa, "Classification of partial 2-D shapes using Fourier descriptors", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. Pami-9, no. 5.pp. 686-690, 1987.

[2] Ro. Jamieson, G. Stephens and S. Kumar, "Fingerprint Identification: An Aid to the Authentication Process" *Information systems control journal,* vol. 1, pp. 1-4, 2005.

[3] Anoop M. Namboodiri, "On-Line Handwritten Document Understanding" PhD dissertation, Michigan State University, 2004.

[4] http://www.electronics-manufacturers.com/info/computers-and-laptops/computer-mouse.html, 2006.

[5] Nick Mokey, http://reviews.digitaltrends.com/first-look/142/iogear-digital-scribe, 2007.

[6] T. Taxt and K.W. Bjerde, "Classification of Handwritten Vector Symbols using Elliptic Fourier Descriptors", University of Bergen. pp. 123-128, 1994.

[7] R. O. Duda and P. E. Hart, "Pattern Classification and Scene Analysis", *IEEE Transaction on Automatic Control,* pp. 462-482, 1973

[8] N. K. Alang-Rashid and A. S. Heger, "A General Purpose Fuzzy Logic Code", *Chemical and Nuclear Engineering Department University of New Mexico*, pp. 733-742, 1992.

[9] G. F. Luger, "Artificial Intelligence Structure and Strategies for Complex Problem Solving", 5th Edition, pp. 353-357, 2005.

[10] D. Ramot, M. Friedman, G. Langholz and A. Kandel, "Complex Fuzzy Logic", *IEEE Transactions on Fuzzy Systems,* vol. 11, no. 4, pp. 450-460, 2003.

[11] L. A. Zadeh, "Fuzzy Logic: Issues, Contentions and Perspectives", Computer Science Division and the Electronics Research Laboratory, University of California, vi-183, 1994.

[12] I. B. Turksen, "Fuzzy Logic: Review of Recent Concerns", Department of Mechanical and Industrial Engineering, University of Toronto, pp. 2975-2978, 1997.

[13] L. A. Zadeh, "Is There a Need for Fuzzy Logic?", Department of Electrical Engineering and Computer Sciences, University of California.

[14] B. Jayasekara, A. Jayasiri and L. Udawatta, "An Evolving Signature Recognition System" *First International Conference on Industrial and Information Systems ICIIS,* pp. 529 -534, 2006.

[15] R. Plamondon and N. Srihari, "On-line and Off-line Handwritten Recognition: A Comprehensive Survey.", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 63-84, January 2000.

[16] A. Zimmer and L. Ling, "A Hybrid On/Off Line Handwritten Signature Verification System", *Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)*, pp. 1-5, 2003.

[17] R. Sabourin, G. Genest, and J. Preteux, "Off-Line Signature Verification by Local Granulometric Size Distributions", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 19, no. 9, pp. 976-987, 1997.

[18] L. Lee, T. Berger, and E. Aviczer, "Reliable On-line Human Signature verification systems", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 18, no. 6, pp. 643- 647, 1996.

[19] Y. Qi and R. Hunt, "Signature verification using global and grid features", *Pattern Recognition*, vol. 27, iss. 12, pp. 1621-1629, 1994.

[20] B. Fang, H. Leung, Y. Tang, W. Tse, K. Kwok, and K. Wong, "Off-line signature verification by the tracking of feature and stroke positions", *Pattern Recognition*, vol. 36, iss. 1, pp. 91-101, 2003.

[21] L. Wan, Z. Lin, and R. Zhao, "Off-line signature verification incorporating the prior model", *In proceedings of International Conference on Machine Learning and Cybernetics*, vol. 3, pp. 1602-1606, 2003.

[22] H. Baltzakis and N. Papamarkos, "A new signature verification technique based on a two stage neural network classifier", *Engineering applications of Artificial Intelligence*, vol. 14, iss. 1, pp.95-103, 2001.

[23] P. Drouhard, R. Sabourin, and M. Godbout, "A neural network approach to off-line signature verification using directional PDF", *Pattern Recognition*, vol. 29, iss. 3, pp. 415-424, 1996.

[24] K. Huang and H. Yan, "Off-line signature verification based on geometric feature extraction and neural network classification", *Pattern Recognition*, vol. 30, iss.1, pp. 9-17, January 1997.

[25] A. Ismail and S. Gad, "Off-line Arabic signature recognition and verification", *Pattern Recognition*, vol. 33, iss. 10, pp.1727-1740, 2000.

[26] L. L. Lee, T. Berger and E. Aviczer, “Reliable On-line Human Signature Verification System”, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 18, no. 6, pp. 643-647, 1996.

[27] R. Plamondon and G. Lorette, ”Automatic Signature Verification and Writer Identification-The State of The Art,” *Pattern Recognition,* vol. 22, no. 2, pp. 107-131,1989.

[28] F. Leclerc and R. Plamondon, ”Automatic Signature Verification: the State of The Art,” *International Journal Pattern Recognition and Artificial Intelligence,* vol. 8, no. 3, pp. 643-660,1994.

[29] A. Zimmer and L. L. Ling, "Preprocessing: Segmenting by Stroke Complexity", *Proceedings of the VI Iber-American Symposium on Pattern Recognition,* pp. 89-94, 2001.

[30] V. S. Nalwa, “Automatic On-line Signature Verification”, *Proceedings of IEEE,* pp. 215-239, 1997.

[31] R. Plamondon and G. Lorette, “Identity Verification from Automatic Processing of Signatures: Bibliography,” *Computer Processing of Handwriting*, pp. 65-85, 1990.

[32] G. Lorette and R. Plamondon, “Dynamic Approaches to Handwritten Signature Verification,” *Computer Processing of Handwriting*, pp. 21-47, 1990.

[33] T. Huang and M. Yasuhara, “Dynamic Programming Matching Applied to Off-line Handwritten Character Recognition”, *IEEE,* pp. 418-423, 1996.

[34] A. E. Jacquin,”Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations”, *IEEE Transaction on Image Processing,* pp.18-30, 1992.

[35] Y. Fisher,"Fractal Image Compression: Theory and Application". Department of Mathematics Technion Isreal Institute of Technology University of California, pp. 1-20, 1992.

[36] P. Temdee, D. Khawparisuth and K. Chamnongthai "Face Recognition by using Fractal Encoding and Backpropagation Neural Network",*15th Information Science, Signal Processing and their Applications*, pp. 159-161,1999.

[37] H. Ebrahimpour, V. Chandran and S. Sridharan "Face Recognition Using Fractal Codes" ,*IEEE*, pp.58-61, 2001.

[38] T.Tan and H.Yan," Face Recognition by Fractal Transformations", *IEEE International Conference on Acoustic, Speech and Signal Processing*, vol.6, pp.3537-3540, 1999.

[39] S. Mozaffari, K. Faez and M. Ziaratban," Character Representation and Recognition Using Quadtree-based Fractal Encoding Scheme" *International Conference on Document Analysis and Recognition,* pp.819-823, 2005.

[40] S. Mozaffari, K. Faez and H. R. Kanan. "Performance Evaluation of Fractal Feature in Recognition of Postal Codes Using an RBF Neural Network and SVM Classifier", MVA, pp. 562-565, 2005.

[41] S. Mozaffari, K. Faez and H. R. Kanan. "Feature Comparison between Fractal Codes and Wavelet Transform in Handwritten Alphanumeric Recognition Using SVM Classifier", *International Conference on Pattern Recognition*, vol.2, pp. 331-334, 2004.

[42] S. Mozaffari, K. Faez and H. R. Kanan. "Recognition of Isolated Handwritten Farsi/Arabic Alphanumeric Using Fractal Codes", SSIAI, pp. 104-108, 2004.

[43] K. Huang and H. Yan,"Signature Verification using Fractal Transformation". *International Conference on Pattern Recognition*, pp.851-854, 2004.

[44] H. Potlapalli.; R.C. Luo, "Fractal-based Classification of Natural Textures", *IEEE Transactions on Industrial Electronics*, vol. 45, Pp.142–150, 1998.

[45] S. Mozaffari, K. Faez and F. Faradji, "One Dimensional Fractal Coder for Online Signature Recognition", *International Conference on Pattern Recognition (ICPR),* pp. 1-4, 2006.

[46] C. C. Tappert, C. Y. Suen, and T. Wakahara, "On-line Handwriting Recognition-A Survey", *IEEE,* pp. 1123- 1132, 1988.

[47] J. Duvernoy and D. Charraut, "Stability and Stationarity of Cursive handwriting", *Pattern Recognition,* vol. 11, pp. 145-154, 1979.

[48] A. M. Wing, "Variability in handwritten characters", Visible Language, vol. 13, pp. 283-298, 1979.

[49] T. T. Kuklinski, "Components of handprint style variability", *Proceedings of 7th International Conference Pattern Recognition,* pp. 924-926, 1984.

[50] J. R. Ward and T. T. Kuklinski, "A Predictive Model for Variability, with Implications for the Design of Online Character Recognition Systems", *IEEE Transaction Management and Cybernetic*, 1987.

[51] H. Arakawa, K. Odaka and I. Masuda, "On-line Recognition of handwritten Characters- Alphanumerics", *Proceedings of 4th International Conferencing Pattern Recognition,* pp. 810-812, 1978.

[52] M. I. Bernstien, "A Method for Recognizing Handprinted Characters in Real-Time", *Pattern Recognition*, pp. 109-114, 1968.

[53] J. M. Hollerbach, "Understanding Manipulator Control by Synthesizing Human Handwriting", *Artificial Intelligence: An MIT Perspective,* vol. 2 pp. 311-332, 1979.

[54] N. M. Herbst and C. N. Liu, "Automatic Signature Verification Based on Accelerometry", *IBM Research and Development,* vol. 21, pp. 245-253, 1977.

[55] N.M. Herbst and C. N. Liu, "Automatic Verification of Signature by Means of Acceleration", *Proceedings of IEEE Computer Science Conference on Pattern Recognition and Image Processing,* pp. 331-336, 1977.

[56] J. Zheng and G. Zhu, "On-line Handwriting Signature Recognition Based on Wavelet Energy Feature Matching", *Proceedings of the 6th World Congress on Intelligent Control and Automation,* pp. 9885-9888, 2006.

[57] A. M. Darwish and G. A. Auda, "A New Composite Feature Vector for Arabic Handwritten Signature Verification," *Proceedings of IEEE International Conference on Acoustics*, vol. 2, pp. 613-616, 1994.

[58] L. L. Lee, T. Berger and E. Aviczer, "Reliable On-line Human Signature Verification Systems", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 18, no. 6, pp. 643-647, 1996.

[59] D. K. R. McCormack, M. B. Brown, and J. F. Pedersen, "Neural Network Signature Verification Using Haar Wavelet and Fourier Transforms," *Proceedings of the SPIE*, vol. 2046, pp. 14-25, 1993.

[60] N. Mohakrishnan, M. J. Paulik, and M. Khalil, "On-line Signature Verification Using a Non-Stationary Autoregressive Model Representation," *IEEE International Symposium on Circuits and Systems*, pp. 2303-2306, 1993.

[61] L. Yang, B. R. Widjaja, and R. Prasad, "Application of Hidden Markov Model for Signature Verification," *Pattern Recognition*, vol. 28, no. 2, pp. 861-870, 1996.

[62] S. D. Connell, "On-line Handwriting Recognition Using Multiple Pattern Class Model", *Phd Thesis, Department of Computer Science, Michigan State University,* 2000.

[63] H. Yan, "Handwritten Digit Recognition Using an Optimizer Nearest Neighbour Classifier", *Pattern Recognition Letters,* vol. 15, no. 2, pp. 207-211, 1994.

[64] B. V. Dasarathy, "Nearest Neighbor Norms: NN Pattern Classification Techniques", *IEEE Computer Society Press,* 1991.

[65] C.-L. Liu and M. Nakagawa, "Evaluation of Prototype Learning Algorithms for Nearest-Neighbour Classifier in Application to handwritten Character Recognition", *Pattern Recognition Society,* vol 34, pp. 601-615, 2001.

[66] C. C. Tappert, C. Y. Suen and T. Wakahara, "State of the Art in Online Handwriting Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 12, no. 8, pp.787-808, 1990.

[67] T. M. Cover and P. E. Hart, "Nearest Neighbour Pattern Classification", *IEEE Transactions on Information Theory,* vol. IT-13, no. 1, pp. 21-27, 1967.

[68] E. H. Ratzlaff, "Inter-Line Distance Estimation and Text Line Extraction for Unconstrained Online Handwriting" *Proceedings of the 6th International workshop on Frontiers in Handwriting Recognition*, pp. 33-42, 2000.

[69] D.S. Zhang, and G.J. Lu, "A comparative Study of Curvature Scale Space and Fourier Descriptors for Shape-based Image Retrieval", *Journal of Visual Communication and Image Representation*, pp. 41-60, 2003.

[70] G. Zhang, Z. M. Ma, Q. Tong, Y. He and T. Zhao, "Shape Feature Extraction Using Fourier Descriptors with Brightness in Content-based Medical Image Retrieval", *IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing,* pp. 71-74, 2008.

[71] R. Kakarala, "Testing for Convexity with Fourier descriptors", *Department of Electrical and Electronic Engineering, University of Auckland,* pp. 1-3.

[72] A. K. Jain, F. D. Griess and S. D. Connell, "On-line Signature Verification", *The Journal of The Pattern Recognition Society,* vol. 35 pp. 2963-2972, 2002.

[73] R. Martens and L. Claesen, "Dynamic programming optimization for on-line signature verification", *Proceedings of International Conference on Document Analysis and Recognition,* pp. 653–656 1997.

[74] H. Sakoe and *S.* Chiba, "Dynamic Programming Optimization for Spoken Word Recognition", *IEEE Transaction on Acoustics Speech and Signal Processing*, vol. 26, no. 1, pp. 43- 49, 1978.

[75] E. J. Keogh and M. J. Pazzani, "Derivative Dynamic Time Warping", Department of Information and Computer Science University of California.

[76] D. J. Berndt and J. Clifford, "Using Dynamic Time Warping to Find Patterns in Time Series", *AAAI-94 Workshop on Knowledge Discovery in Databases,* pp. 359-370, 1994.

[77] W. Shang, H. Huang, H. Zhu, Y. Lin, Z. Wang and Y. Qu, "An Improved kNN Algorithm-Fuzzy kNN", *Computational Intelligence and Security,*v. 3801 pp. 741-746, 2005.

[78] T. M. Cover and P. E. Hart, "Nearest Neighbor Pattern Classification", *IEEE Transaction. Information. Theory,* v. IT-13 pp 21-27, 1967.

[79] D. Driankov, H. Hellendoorn and M. Reinfrank, "An introduction To Fuzzy Control" pp. 38-74, 1993.

[80] S. Theodoridis and K. Koutroumbas, "Pattern Recognition" pp 44-46, 2003.

[81] L. Devroye, L. Gyorfi and G. Lugosi, "A Probabilistic Theory of Pattern Recognition", 1996.

[82] D. Rafiei and A. O. Mendelzon, "Efficient Retrieval of Similar Shapes", *The VLDB Journal manuscript,* pp. 1-12.

[83] M. Jamshidi, N. Vadiee and T. J. Ross, "Fuzzy Logic and Control Software and Hardware Applications" *Environmental and Itelligent Manufacturing Systems Series,* pp. 10-35

[84] Sad L. I. Kuncheva, "Fuzzy Classifier Design" *Studies in Fuzziness and Soft Computing,* pp. 117-155.

[85] H. Hamilton, E. Gurak, L. Findlater, W. Olive and J. Ranson, http://www2.cs.uregina.ca/~hamilton/courses/831/notes/confusion_matrix/confusion_matrix.html

[86] A. B. Tickle, R. Andrews, M. Golea, and J. Diederich, "Rule Extraction from Trained Artificial Neutral Networks" *Neural Network Analysis, Architectures and Applications,* pp. 61-99, 1997.

[87] A. K. Jain, and J. Mao, "Special Issue on Artificial Neural Networks and Statistical Pattern Recognition", *IEEE Transactions on Neural Networks,* pp. 1-3, 1997.

[88] R. P. W. Duin "A Note on Comparing Classifier", *Pattern Recognition Letters,* pp. 529-536, 1996.

[89] A. Folkers and H. Samet, "Content-based Image Retrieval Using Fourier Descriptors on a Logo Database", *Proceedings of the 16th International Conference on Pattern Recognition,* pp. 521-524, 2002.

[90] A. K. Jain. "Fundamentals of Digital Image Processing" Information and Systems Science Series. Prentice Hall, 1989.